

TRABAJO FIN DE GRADO



UNIVERSIDAD CARLOS III DE MADRID

Sistema de gestión de currículos

Titulación: Grado en Ingeniería en Informática

Autor: Sergio Jiménez Yébenes

Tutor: Antonio Berlanga de Jesús

Resumen

El proyecto desarrollado se corresponde con un sistema de gestión de CVS. El objetivo de este sistema es el de poder tratar los datos de los CVS, de tal manera que se puedan modificar o eliminar esos datos, e incluso añadir información nueva. Por otro lado, se podrá utilizar esa información para realizar un seguimiento sobre cada CV, así como poder exportarlo a un formato específico. Además, se pueden crear CVS nuevos desde cero.

La idea de la creación de este sistema reside en un problema común que tienen muchas empresas. Cada empresa, tiene un apartado de selección en el cual se dedican a elegir candidatos para ofrecerles trabajos, cursos, u otras ofertas. Estos candidatos, mayormente, suelen ser buscados en portales web dedicados al envío de CVS a las empresas que realizan ofertas, como por ejemplo Infojobs o Tecnoempleo. Todos los portales web, tienen un formato específico con el cual envían los CVS, y eso a la hora de querer tratarlos por igual, puede causar muchos problemas. Además, muchos CVS tienen excesiva información, o por otro lado, cuando una persona de selección de una empresa se pone en contacto con una persona por haber visto su CV, es posible que durante la conversación, decidan añadir al CV nueva información.

Este sistema es capaz de unificar todo lo anterior. El sistema será capaz de recibir CVS de distintos formatos, guardarlos, y podrá modificar en todo momento esa información. Se podrán hacer cargas masivas de CVS, y posteriormente buscarlos para poder tratarlos independientemente. Cada CV podrá ser exportado a un formato específico, y se le podrá incluir un seguimiento, esto es, si se ha llamado al candidato, si se le va a conceder una entrevista...etc. También, se podrán crear CVS a mano desde cero.

Contenido

1. Acrónimos.....	15
2. Introducción.....	16
2.1. Motivación	16
2.2. Objetivos	17
2.3. Medios.....	17
Medios hardware	17
Medios software	18
3. Estado del arte	20
3.1. Sistemas de gestión documental	20
3.2. Text mining.....	22
3.3. Sistemas de bases de datos.....	23
3.4. SQL vs NO-SQL.....	23
4. Análisis	25
4.1. Obtención de CVS.....	25
4.2. Base de datos	28
4.3. Programa de recogida de información.....	32
4.4. Página web	33
4.5. Requisitos del sistema.....	35
Requisitos funcionales.....	35
Requisitos no funcionales	50
4.6. Casos de uso	51
Casos de uso	52
Diagramas de caso de uso.....	61
5. Desarrollo	63
5.1. Base de datos	63
Conocimientos_mantenimiento	64
Contacto	64
Idiomas_general.....	65

Países.....	65
Perfiles.....	66
Provincias	66
Titulaciones	67
Usuarios.....	67
Datos_personales.....	68
Otros_datospersonales	69
Estudios	70
Otros_titulos	71
Otros_cursos	71
Experiencia	72
Conocimientos_tecnologias	73
Idiomas.....	73
Otros_datos.....	74
Futuro_empleo.....	75
Peticion.....	76
Conocimientos_peticion	77
Idiomas_peticion.....	77
CV_peticion	78
Seguimiento	79
Análisis.....	79
5.2. Programa de recogida de información.....	82
5.3. Página web	88
Index.....	89
Cabeceras	93
Alta candidato	94
Búsqueda candidato.....	100
6. Resultado.....	101
7. Conclusiones	119

7.1. Trabajos futuros	120
8. Bibliografía	121
9. Anexos.....	122
9.1. Planificación	122
Análisis.....	122
Diseño.....	122
Implementación y pruebas.....	122
Diagrama de Gantt	123
9.2. Presupuesto	125
Coste de material Hardware	125
Coste de material Software.....	125
Coste de mano de obra	126
Coste total del proyecto	126
9.3. Responsabilidad jurídica	126
Infojobs.....	126
Tecnoempleo.....	127
Responsabilidad jurídica del SGCV	128
10. Habilidad en Inglés.....	129
Introduction	129
Abstract	130
State of the art	130
Analysis.....	131
Development.....	134
Result.....	136
Planning and budgeting.....	138
Conclusions	139
Future work	139

Ilustración 1: Medio software XAMPP	18
Ilustración 2: Medio software Notepad++	18
Ilustración 3: Medio software Microsoft SQL Server 2012	19
Ilustración 4: Medio software Mozilla firefox.....	19
Ilustración 5: Sistemas de gestión documental	20
Ilustración 6: Text mining.....	22
Ilustración 7: Sistemas de bases de datos.....	23
Ilustración 8: SQL vs NO-SQL.....	23
Ilustración 9: Obtención de CVS -1.....	25
Ilustración 10: Obtención CVS -2.....	26
Ilustración 11: Obtención de CVS -3.....	26
Ilustración 12: Obtención de CVS -4.....	27
Ilustración 13: ODBC-1	28
Ilustración 14: ODBC-2	28
Ilustración 15: ODBC-3	29
Ilustración 16: ODBC-4	29
Ilustración 17: ODBC-5	30
Ilustración 18: ODBC-6	30
Ilustración 19: Ejemplo creación Base de datos.....	31
Ilustración 20: Ejemplo creación tabla.....	31
Ilustración 21: Lectura de CVS para guardar	32
Ilustración 22: Estructura del programa	32
Ilustración 23: Ejemplo de html	33
Ilustración 24: Ejemplo de javaScript.....	33
Ilustración 25: Ejemplo de CSS	33
Ilustración 26: Ejemplo de PHP	33
Ilustración 27: Ejemplo de página web de la aplicación	34
Ilustración 28: Usuario "Administrador"	61
Ilustración 29: Usuario "Petición"	62

Ilustración 30: Usuario "Gestor"	62
Ilustración 31: Usuario "Usuario externo"	62
Ilustración 32: Base de datos	63
Ilustración 33: Función NewGuid().....	63
Ilustración 34: Tabla Conocimientos_mantenimiento.....	64
Ilustración 35: Tabla Contacto	64
Ilustración 36: Tabla Idiomas_general	65
Ilustración 37: Tabla Paises	65
Ilustración 38: Tabla Perfiles	66
Ilustración 39: Tabla Provincias.....	66
Ilustración 40: Tabla Titulaciones.....	67
Ilustración 41: Tabla Usuarios	67
Ilustración 42: Tabla Datos_personales	68
Ilustración 43: Ejemplo de relación entre tablas	69
Ilustración 44: Tabla Otros_datospersonales.....	69
Ilustración 45: Tabla Estudios.....	70
Ilustración 46: Tabla Otros_titulos.....	71
Ilustración 47: Tabla Otros_cursos.....	71
Ilustración 48: Tabla Experiencia	72
Ilustración 49: Tabla Conocimientos_tecnologias	73
Ilustración 50: Tabla Idiomas	73
Ilustración 51: Tabla Otros_datos	74
Ilustración 52: Tabla Futuro_empleo	75
Ilustración 53: Tabla Peticion	76
Ilustración 54: Tabla Conocimientos_peticion.....	77
Ilustración 55: Tabla Idiomas_peticion	77
Ilustración 56: Tabla CV_peticion.....	78
Ilustración 57: Tabla Seguimiento.....	79
Ilustración 58: Ejemplo de select	80

Ilustración 59: Ejemplo de inserción de conocimientos	80
Ilustración 60: Ejemplo de importar archivos	82
Ilustración 61: Función comprobarFormato()	82
Ilustración 62: Ejemplo recogida datos personales de infojobs	83
Ilustración 63: Ejemplo de estudios de infojobs	84
Ilustración 64: Ejemplo de recursividad en estudios de infojobs	84
Ilustración 65: Variable conn.....	85
Ilustración 66: Ejemplo de includes	86
Ilustración 67: Función conectarBD()	86
Ilustración 68: Valores globals	86
Ilustración 69: Ejemplo de inserción en base de datos.....	87
Ilustración 70: Control de CV repetidos	87
Ilustración 71: Pantalla de Index	89
Ilustración 72: Crear usuario	89
Ilustración 73: Método cifrado.....	90
Ilustración 74: Comprobar contraseña	90
Ilustración 75: Perfiles de usuario.....	90
Ilustración 76: Inicio Administrador	91
Ilustración 77: Inicio Usuario externo	91
Ilustración 78: Ejemplo de control perfiles	92
Ilustración 79: Comprobar perfil en Mantenimiento.php	93
Ilustración 80: Cabecera de páginas	93
Ilustración 81: Html de cabecera de páginas	93
Ilustración 82: JavaScript de cabecera de páginas.....	93
Ilustración 83: Pantalla de alta candidato.....	94
Ilustración 84: Html alta candidato	94
Ilustración 85: CSS de alta de candidato	95
Ilustración 86: Html Estudios	96
Ilustración 87: Funcion ocultarEstudios().....	96

Ilustración 88: Botón añadir Estudios	96
Ilustración 89: Clase masestudio.....	96
Ilustración 90: Función addEstudio()	97
Ilustración 91: Función borrarEstudio()	97
Ilustración 92: Guardar cambios alta candidato	98
Ilustración 93: Pop-up de guardar cambios	98
Ilustración 94: Form alta candidato	98
Ilustración 95: Inserción de Otros_titulos en alta candidato	99
Ilustración 96: Inserción correcta de alta candidato.....	99
Ilustración 97: Resultado búsqueda candidato.....	100
Ilustración 98: Resultado Index.....	101
Ilustración 99: Resultado Inicio	102
Ilustración 100: Resultado Alta	102
Ilustración 101: Resultado prueba de alta candidato formulario	103
Ilustración 102: Resultado alta de candidato formulario.....	103
Ilustración 103: Resultado Importar CV-1.....	104
Ilustración 104: Resultado Importar CV-2.....	104
Ilustración 105: Resultado Importar CV-3.....	105
Ilustración 106: Resultado búsqueda-1.....	105
Ilustración 107: Resultado búsqueda-2.....	106
Ilustración 108: Resultado mostrarCurrículum	106
Ilustración 109: Resultado exportar a word-1	107
Ilustración 110: Resultado exportar a word-2	107
Ilustración 111: Resultado Edición CV.....	108
Ilustración 112: Resultado Edición Guardar cambios-1	108
Ilustración 113: Resultado Edición Guardar cambios-2	109
Ilustración 114: Resultado comprobación edición-1	109
Ilustración 115: Resultado comprobación edición-2	110
Ilustración 116: Resultado borrar candidato-1	110

Ilustración 117: Resultado borrar candidato-2	111
Ilustración 118: Resultado confirmar borrar candidato.....	111
Ilustración 119: Resultado peticiones	112
Ilustración 120: Resultado Alta petición	112
Ilustración 121: Resultado búsqueda peticiones	113
Ilustración 122: Resultado mostrar petición.....	113
Ilustración 123: Resultado gestión peticiones-1	114
Ilustración 124: Resultado agregar CV a petición	114
Ilustración 125: Resultado gestión peticiones-2	115
Ilustración 126: Resultado verCV asociado	115
Ilustración 127: Resultado crear seguimiento	116
Ilustración 128: Resultado eliminar CV asociado-1.....	116
Ilustración 129: Resultado eliminar CV asociado-2.....	117
Ilustración 130: Resultado mantenimiento.....	117
Ilustración 131: Resultado mantenimiento conocimientos	118
Ilustración 132: Importar varios CVS-1	118
Ilustración 133: Importar varios CVS-2	119
Ilustración 134: Diagrama de Gantt	124
Ilustración 135: State of the art	130
Ilustración 136: Obtaining CVS.....	131
Ilustración 137: Database.....	131
Ilustración 138: Information collection program	132
Ilustración 139: Web page	132
Ilustración 140: User "Administrator"	134
Ilustración 141: Upload CV	136
Ilustración 142: Example of result search	137
Ilustración 143: Gantt diagram	138
Ilustración 144: Total budget	139

Tabla 1: Medios hardware - PC	17
Tabla 2: Tabla tipo de requisitos	35
Tabla 3: Requisito funcional de funcionamiento FF01.....	35
Tabla 4: Requisito funcional de funcionamiento FF02.....	36
Tabla 5: Requisito funcional de funcionamiento FF03.....	36
Tabla 6: Requisito funcional de funcionamiento FF04.....	36
Tabla 7: Requisito funcional de funcionamiento FF05.....	36
Tabla 8: Requisito funcional de funcionamiento FF06.....	36
Tabla 9: Requisito funcional de funcionamiento FF07.....	37
Tabla 10: Requisito funcional de funcionamiento FF08.....	37
Tabla 11: Requisito funcional de funcionamiento FF09.....	37
Tabla 12: Requisito funcional de funcionamiento FF10.....	37
Tabla 13: Requisito funcional de funcionamiento FF11.....	37
Tabla 14: Requisito funcional de funcionamiento FF12.....	38
Tabla 15: Requisito funcional de funcionamiento FF13.....	38
Tabla 16: Requisito funcional de funcionamiento FF14.....	38
Tabla 17: Requisito funcional de funcionamiento FF15.....	38
Tabla 18: Requisito funcional de funcionamiento FF16.....	38
Tabla 19: Requisito funcional de funcionamiento FF17.....	39
Tabla 20: Requisito funcional de funcionamiento FF18.....	39
Tabla 21: Requisito funcional de funcionamiento FF19.....	39
Tabla 22: Requisito funcional de funcionamiento FF20.....	39
Tabla 23: Requisito funcional de funcionamiento FF21.....	40
Tabla 24: Requisito funcional de funcionamiento FF22.....	40
Tabla 25: Requisito funcional de funcionamiento FF23.....	40
Tabla 26: Requisito funcional de funcionamiento FF24.....	40
Tabla 27: Requisito funcional de funcionamiento FF25.....	41
Tabla 28: Requisito funcional de funcionamiento FF26.....	41

Tabla 29: Requisito funcional de funcionamiento FF27.....	41
Tabla 30: Requisito funcional de interfaz FI01.....	41
Tabla 31: Requisito funcional de interfaz FI02.....	41
Tabla 32: Requisito funcional de interfaz FI03.....	42
Tabla 33: Requisito funcional de interfaz FI04.....	42
Tabla 34: Requisito funcional de interfaz FI05.....	42
Tabla 35: Requisito funcional de interfaz FI06.....	42
Tabla 36: Requisito funcional de interfaz FI07.....	43
Tabla 37: Requisito funcional de interfaz FI08.....	43
Tabla 38: Requisito funcional de interfaz FI09.....	43
Tabla 39: Requisito funcional de interfaz FI10.....	43
Tabla 40: Requisito funcional de interfaz FI11.....	44
Tabla 41: Requisito funcional de interfaz FI12.....	44
Tabla 42: Requisito funcional de interfaz FI13.....	44
Tabla 43: Requisito funcional de interfaz FI14.....	44
Tabla 44: Requisito funcional de interfaz FI15.....	45
Tabla 45: Requisito funcional de interfaz FI16.....	45
Tabla 46: Requisito funcional de interfaz FI17.....	45
Tabla 47: Requisito funcional de interfaz FI18.....	45
Tabla 48: Requisito funcional de interfaz FI19.....	46
Tabla 49: Requisito funcional de base de datos FB01.....	46
Tabla 50: Requisito funcional de base de datos FB02.....	46
Tabla 51: Requisito funcional de base de datos FB03.....	46
Tabla 52: Requisito funcional de base de datos FB04.....	47
Tabla 53: Requisito funcional de base de datos FB05.....	47
Tabla 54: Requisito funcional de base de datos FB06.....	47
Tabla 55: Requisito funcional de base de datos FB07.....	47
Tabla 56: Requisito funcional de base de datos FB08.....	48
Tabla 57: Requisito funcional de base de datos FB09.....	48

Tabla 58: Requisito funcional de base de datos FB10.....	48
Tabla 59: Requisito funcional de base de datos FB11.....	48
Tabla 60: Requisito funcional de base de datos FB12.....	49
Tabla 61: Requisito funcional de base de datos FB13.....	49
Tabla 62: Requisito funcional de base de datos FB14.....	49
Tabla 63: Requisito funcional de base de datos FB15.....	49
Tabla 64: Requisito funcional de base de datos FB16.....	50
Tabla 65: Requisito no funcional NF01.....	50
Tabla 66: Requisito no funcional NF02.....	50
Tabla 67: Requisito no funcional NF03.....	50
Tabla 68: Requisito no funcional NF04.....	51
Tabla 69: Requisito no funcional NF05.....	51
Tabla 70: Tabla tipo de casos de uso.....	51
Tabla 71: Caso de uso CU01	52
Tabla 72: Caso de uso CU02	52
Tabla 73: Caso de uso CU03	53
Tabla 74: Caso de uso CU04	53
Tabla 75: Caso de uso CU05	54
Tabla 76: Caso de uso CU06	54
Tabla 77: Caso de uso CU07	55
Tabla 78: Caso de uso CU08	55
Tabla 79: Caso de uso CU09	56
Tabla 80: Caso de uso CU10	56
Tabla 81: Caso de uso CU11	57
Tabla 82: Caso de uso CU12	57
Tabla 83: Caso de uso CU13	58
Tabla 84: Caso de uso CU14	58
Tabla 85: Caso de uso CU15	59
Tabla 86: Caso de uso CU16	59

Tabla 87: Caso de uso CU17	60
Tabla 88: Caso de uso CU18	60
Tabla 89: Actividades de planificación	123
Tabla 90: Duración total del proyecto.....	124
Tabla 91: Presupuesto material hardware.....	125
Tabla 92: Presupuesto material software	125
Tabla 93: Presupuesto mano de obra	126
Tabla 94: Presupuesto coste total del proyecto	126
Tabla 95: Example requirement.....	133
Tabla 96: Planning activities.....	138

1. Acrónimos

- **SGCV**: Sistema de gestión de currículos (este proyecto).
- **CV**: Currículo.
- **CVS**: Currículos.
- **GB**: Gigabyte, unidad de almacenamiento de información.
- **HW**: Hardware.
- **SW**: Software.
- **CPU**: Unidad central de procesamiento.
- **RAM**: Random Access Memory (memoria de acceso aleatorio).
- **XAMPP**: El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl.
- **SQL**: Structured Query Language (lenguaje de consulta estructurado).
- **PHP**: Hypertext Preprocessor (procesador de hipertexto).
- **HTML**: HyperText Markup Language (lenguaje de marcas de hipertexto).
- **CSS**: Cascading Style Sheets (hojas de estilo en cascada).
- **TXT**: Archivo de texto simple.
- **GUID**: identificador único global.
- **NO-SQL**: Not only SQL (No solo SQL).
- **DHT**: Tablas hash distribuidas.
- **P2p**: Peer-to-peer (red de pares).
- **ODBC**: open database connectivity (conectividad abierta de base de datos).
- **BOE**: Boletín oficial del estado.
- **LOPD**: Ley orgánica de protección de datos.

2. Introducción

En la actualidad, nos encontramos con que la tecnología abarca prácticamente todo el entorno social que nos rodea. A la hora de encontrar trabajo, las personas realizan un CV donde exponen su información personal, sus estudios, su experiencia profesional...con la finalidad de hacer llegar los CVS a empresas que pudieran estar interesadas en contratarles.

Hasta hace no mucho, estos CVS eran entregados en mano a las empresas o particulares que realizaban una oferta o incluso sin la necesidad de realizar una oferta. No obstante, con el impulso de las nuevas tecnologías, es decir, internet, ordenadores, teléfonos móviles...estos CVS son enviados a través de internet, frecuentemente por correo electrónico.

En este documento voy a explicar con detalle la creación de un sistema capaz de tratar la información de todos esos CVS enviados por internet con una finalidad concreta.

2.1. Motivación

La gestión de CVS es un tema muy importante, ya que la mayoría de los trabajos son ofrecidos a gente con un buen CV, presentable y vistoso. Este sistema ayuda en el proceso de selección de esos CVS que figuran en algunos portales web muy utilizados.

Las empresas disponen de un apartado de selección, en el cual se dedican a ver CVS que cumplan ciertos requisitos y a ponerse en contacto con las personas de esos CVS. Las personas encargadas de realizar este proceso, pueden estar interesadas en mandar esos mismos CVS a otras empresas o a otros sectores que pudieran estar interesados en ese CV, o bien quedárselo para sí mismos para poder gestionarlo a su antojo. Este sistema tendrá que facilitar todo este proceso.

La clave de todo es hacer llegar los CVS a destinos interesados en ellos, además de poder visualizar la información del CV de manera clara y limpia y poder realizar modificaciones. Un CV es una de las principales vías a la hora de encontrar trabajo, cosa que en la mayoría de las personas es una prioridad fundamental. Esta es la principal motivación a la hora de crear este sistema, facilitar y agilizar la selección de candidatos a un puesto de trabajo.

2.2. Objetivos

El objetivo de este proyecto es el de realizar un sistema capaz de poder almacenar CVS, poder gestionarlos y exportarlos a un formato concreto.

Se quiere mostrar como el sistema puede almacenar CVS en la base de datos, ya sean creados manualmente o cogidos desde portales web. Se hará un análisis en profundidad de cómo se realiza el proceso de alta de CVS.

Por otro lado, se va a describir con detalle las posibilidades de gestión del CV una vez guardado (modificación de la información, asociación de CVS con peticiones, seguimiento de CV, entre otros). También se podrán crear peticiones, las que tendrán una serie de requisitos y a las que se les podrá asignar varios CVS siempre que cumplan todos los requisitos de la petición.

Finalmente, probaremos como un CV puede ser exportado a varios formatos diferentes previamente creados y definidos.

2.3. Medios

Los medios utilizados para realizar el sistema los he dividido en medios hardware y medios software, y son los siguientes:

Medios hardware

En primer lugar está el ordenador donde se ha programado el sistema y se han realizado todas las pruebas.

Procesador	Intel® Core™ i5-2450M
CPU	2.50 Ghz
Memoria RAM	8.00 GB
Disco duro	750 GB
Sistema operativo	Windows 7 Home Premium (64 bits)

Tabla 1: Medios hardware - PC

Medios software

Los medios SW se pueden ver a continuación:

1. XAMPP:



Ilustración 1: Medio software XAMPP

XAMPP es un software libre que actúa como servidor independiente de plataforma. Permite hacer gestión de base de datos con MySQL, incluye el servidor web Apache, y los intérpretes para lenguajes de script: PHP y PERL. En este caso, el lenguaje utilizado ha sido PHP, y es un lenguaje que se incluye con html, por lo que todo se ha combinado para crear la página web que sirviera como soporte a la hora de gestionar los CVS. XAMPP crea como un servidor local independiente, y se instala una carpeta "xampp" en el disco duro. Cuando lanzamos el programa, dentro de la carpeta "xampp" hay una carpeta llamada "htdocs", en la cual podemos incluir tantos scripts como queramos. Posteriormente, podemos acceder a ejecutar esos scripts a través del navegador de la siguiente manera:

Localhost:8080/prueba.php

Con localhost indicamos que accedemos al servidor local que crea XAMPP, y 8080 es el puerto del ordenador donde se lanza el servidor. Después, prueba.php es un ejemplo de script que se podría ejecutar. El resultado se puede ver en el navegador.

2. Notepad++:



Ilustración 2: Medio software Notepad++

Notepad++ es un editor de textos que da soporte a muchos lenguajes de programación. Reconoce PHP, así como JavaScript o html, que ha sido lo que se ha utilizado para el proyecto.

3. Microsoft SQL Server 2012:



Ilustración 3: Medio software Microsoft SQL Server 2012

Microsoft SQL Server es un sistema de manejo de bases de datos del modelo relacional. Pese a que el XAMPP permite manejo de base de datos con MySQL, decidí utilizar SQL Server por más comodidad.

4. Navegador:



Ilustración 4: Medio software Mozilla firefox

Para poder ejecutar el sistema, ha sido necesario un navegador web. Se podría usar cualquiera, pero el utilizado para realizar todo el proyecto ha sido Mozilla Firefox. Este navegador además incluye una extensión denominada "Firebug" que ha sido creada y diseñada especialmente para programadores y desarrolladores web. Esta extensión permite ver en el navegador los errores que van surgiendo, los datos que se envían entre scripts, lo que se recibe...y todo tipo de utilidades.

3. Estado del arte

En esta sociedad cada vez más tecnológica en la que vivimos, la búsqueda de un empleo es una razón fundamental en la vida de todas las personas. Para ello, disponer de un buen CV es tarea fundamental para conseguir ese ansiado objetivo. Para realizar el SGCV, ha sido necesario la creación de varios elementos y el empleo de varias técnicas y conceptos. Estos son los siguientes:

3.1. Sistemas de gestión documental

En primer lugar, vamos a definir lo que es un sistema de gestión documental:



Ilustración 5: Sistemas de gestión documental

"Los sistemas de gestión documental son programas de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos científicos, culturales y técnicos. Estos sistemas difieren en aspectos fundamentales de los de gestión de bases de datos convencionales, o de aplicación general, que se utilizan para la gestión de documentos administrativos". [1]

En resumidas cuentas, un sistema de gestión documental es un programa que permite el manejo de una gran cantidad de documentos, esto es, poder almacenarlos en una base de datos, poder acceder a ellos fácilmente y poder tratarlos como se desee. En este caso, el SGCV es un caso particular de sistema documental, en concreto un tipo de software de gestión documental.

"Un software de gestión documental es una aplicación informática que permite el manejo, gestión, conservación, publicación y trabajo sobre documentos electrónicos". [2]

Un software de gestión documental no es más que la aplicación que se encarga de realizar todas las funcionalidades de un sistema de gestión documental. Eso es el SGCV.

En la actualidad, cada vez se ven menos CVS en papel, y prácticamente todos son en formato digital. Los CVS de las personas que buscan empleo circulan por internet hasta llegar a su destinatario, que en caso de considerarlo oportuno, decidirá llamar a la persona identificada en ese CV. Se ha dejado prácticamente de lado el llevar tu CV en mano a los lugares donde se pretende encontrar empleo, y se está llevando a cabo el reparto de tu CV mediante internet,

ya sea por correo electrónico, o por la inscripción en algún portal web dedicado especialmente a CVS.

Precisamente son estos portales web los que monopolizan hoy en día el envío de CVS y por tanto controlan el contacto de las personas con las empresas o particulares que ofrecen un empleo. La importancia que tienen estos portales web por tanto es muy elevada, ya que mucha gente depende de la gestión de estos portales web para que su CV sea visto por otras personas o empresas, y por tanto, dependen de ello para que puedan ponerse en contacto con ellos y realizar una entrevista de trabajo.

Hay muchos portales web dedicados al manejo de CVS. Entre los más conocidos se encuentran por ejemplo Infojobs, Tecnoempleo, Infoempleo, Laboris, Linkedin o eProwin, entre otros. Todos estos portales te permiten crear tu CV, buscar ofertas de trabajo, recoger información sobre empresas, y multitud de funciones muy útiles.

Este proyecto se encarga de alguna manera de unificar todos estos portales web a través del elemento fundamental que comparten: los CVS. El SGCV podrá albergar los CVS de todos estos portales (en este proyecto solo de dos portales, Infojobs y Tecnoempleo) para posteriormente poder hacer una gestión específica sobre ellos. No tengo constancia de si existe alguna aplicación parecida a esta, pues no he encontrado nada parecido. Sin embargo, es posible que de manera interna, alguna empresa disponga de una aplicación similar para la gestión de sus candidatos pero que no esté mostrada al público.

Digamos que prácticamente todos los portales web permiten crear un usuario o varios, rellenar un formulario para crear un CV, enviarlo a las empresas, buscar ofertas... y el mismo portal web tiene un acceso para empresas donde los seleccionadores de candidatos reciben esos CVS y se encargan de estudiarlos, o pueden realizar filtros de búsqueda para obtener CVS con unas características concretas. Sin embargo, ningún portal ofrece la gestión de CVS de otros portales. En este proyecto, además de dar posibilidad de crear un CV rellenando un formulario y realizar una serie de gestiones específicas, se da la posibilidad de almacenar CVS de varios portales web de manera automática.

Existen numerosas ventajas a la hora de utilizar sistemas documentales: [3]

- Mayor eficiencia y productividad: fácil acceso a la documentación y ahorro de tiempo.
- Reducción de costes de archivado de documentos: menos papel.
- Seguridad de datos fiable y cumplimiento normativo automatizado: se puede controlar el acceso a la información y aplicar medidas de seguridad.
- Continuidad empresarial: posibilidad de guardar la información en la nube o en copias para conservarla siempre.

3.2. Text mining

En primer lugar hay que tener claro lo que es Data mining.



Ilustración 6: Text mining

"Data mining es el proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior". [4]

Este concepto surgió para ayudar a la comprensión de los contenidos de una base de datos, y trata de adquirir conocimiento de unos datos originales. Para el Data mining los datos son la materia prima, a los que los usuarios le dan significado para convertirlos en información y que posteriormente los especialistas lo conviertan en conocimiento.

Una vez explicado lo que es Data mining, nos centramos en Text mining, que en lugar de tratar los datos de las bases de datos, se encarga de manejar documentos y textos de organizaciones, administraciones...etc.

"La minería de texto se refiere al examen de una colección de documentos y el descubrimiento de información no contenida en ningún documento individual de la colección; en otras palabras, trata de obtener información sin haber partido de algo tratando de encontrar patrones ocultos". [4]

Por tanto, el Text mining se compone de tres tareas principales, que son:

- Recuperación de información, es decir, seleccionar los textos pertinentes.
- Extracción de la información incluida en esos textos: hechos, acontecimientos, datos clave, relaciones entre ellos, etc.
- Por último se realizaría lo que antes definíamos como Data mining para encontrar asociaciones entre esos datos claves previamente extraídos de entre los textos.

Para la realización del SGCV se han utilizado estos conceptos. Text mining en cuanto a recopilar la información de los CVS (documentos/textos), seleccionando los CVS pertinentes y extrayendo información como datos comunes entre ellos. Por otro lado Data mining, se ha creado una base de datos para guardar la información de estos CVS, que son los datos extraídos de los textos.

3.3. Sistemas de bases de datos

En primer lugar, vamos a definir lo que es un sistema de bases de datos:

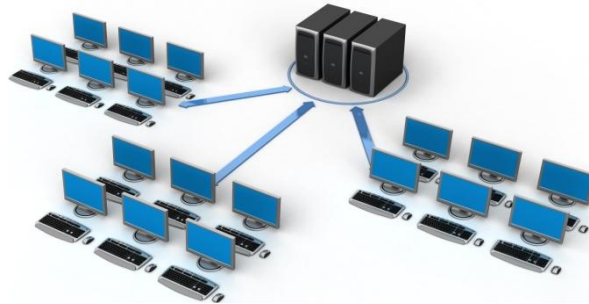


Ilustración 7: Sistemas de bases de datos

"Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos". [5]

Un sistema de base de datos permite almacenar, modificar y extraer grandes cantidades de información que tienen una determinada relación entre sí. Para el caso particular del SGCV, ha sido necesario disponer de un sistema de base de datos, en el cual se creara una base de datos específica, con sus tablas, relaciones y demás componentes, donde poder almacenar toda la información de los CVS.

Además, se ha diseñado e implementado una página web, la cual permite el manejo de toda esa información de la base de datos. Se podrán importar CVS a la base de datos, modificar su contenido, eliminarlo, crear CVS manualmente...etc.

3.4. SQL vs NO-SQL

Aquí surge una cuestión, y es por qué la elección de un modelo de base de datos relacional (SQL) frente a otro no relacional (NO-SQL).



Ilustración 8: SQL vs NO-SQL

A continuación se van a explicar las características de cada modelo:

Características SQL: [6]

- Una base de datos se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni el mismo registro.
- Cada tabla es un conjunto de registros, es decir, de filas y columnas con valores.

- Las relaciones entre tablas se llevan a cabo mediante claves primarias y ajenas.
- Una clave primaria es la clave principal de un registro de una tabla y debe cumplir con la integridad de los datos.
- Las claves ajenas hacen referencia a la clave primaria de otra tabla tomando su valor. De esta manera se hacen las formas relacionales.
- Sistemas disponibles: Microsoft SQL Server, MySQL, Oracle, PostgreSQL, entre otros.

Características NO-SQL: [6]

- Consistencia eventual: no se implementan mecanismos rígidos de consistencia como los presentes en bases de datos relaciones, donde la confirmación de un cambio implica una comunicación del mismo a todos los nodos que lo repliquen. Esta flexibilidad hace que la consistencia se dé, eventualmente, cuando no se hayan modificado los datos durante un periodo de tiempo.
- Estructura distribuida: generalmente se distribuyen los datos mediante mecanismos de DHT, ya que realmente se trata de redes p2p.
- Escalabilidad horizontal: la implementación típica se realiza en muchos nodos de capacidad de procesamiento limitado, en vez de utilizar grandes Mainframes (computadoras centrales).
- Tolerancia a fallos y Redundancia.
- Sistemas disponibles: MongoDB, Cassandra, HBase, Redis, Hadoop, entre otros. [7]

En este caso, necesitamos un modelo que permita guardar la información relacionada entre sí. El SGCV se va a encargar de guardar la información personal de una determinada persona en concreto (de su CV), y posteriormente, relacionará toda la demás información con esa persona, es decir, sus estudios, sus experiencias, sus conocimientos...etc. En caso de haber un cambio para esa persona, necesariamente todas las tablas afectadas por esa persona deben actualizarse con la información modificada. En otras palabras, que es necesario tener una consistencia en todo momento con los datos, y eso lo proporciona las bases de datos relaciones. Además, no se puede permitir la tolerancia a fallos ni la redundancia, pues los datos de las personas son los que son, tienen que estar correctos y no pueden estar duplicados. Si se busca a una persona, esa persona es única. Los modelos de base de datos relacional no permiten tolerancia a fallos ni redundancia, por lo que parece claro que la elección para realizar el SGCV es la de un modelo de base de datos relacional. Por comodidad y mejores conocimientos al respecto, se ha empleado Microsoft SQL Server.

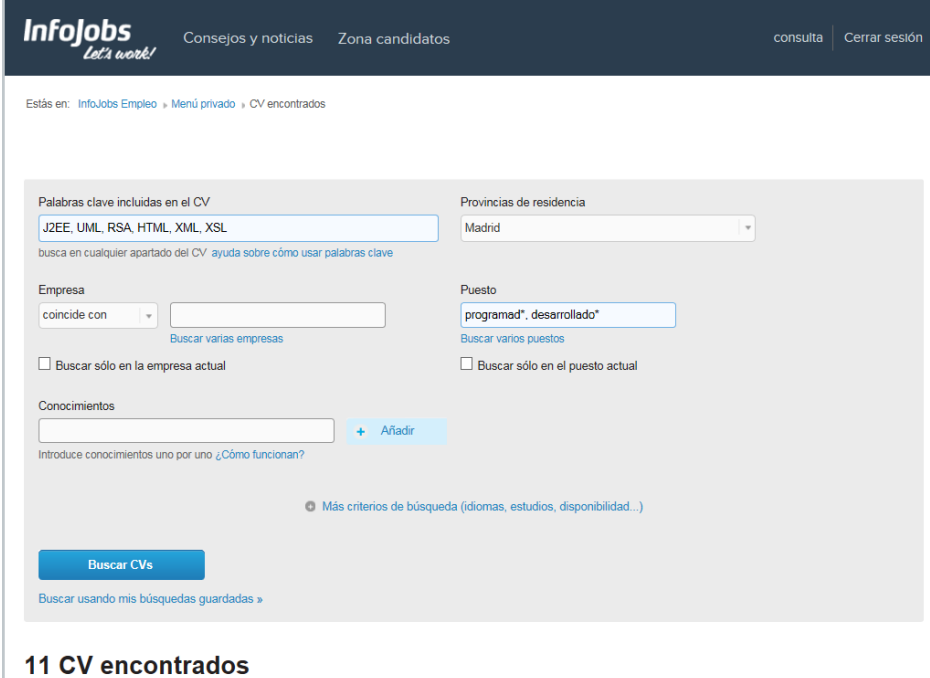
4. Análisis

En este apartado se va a exponer un análisis en profundidad a la hora de afrontar la realización del proyecto.

4.1. Obtención de CVS

En primer lugar y lo más importante, es saber cómo son los CVS que se obtienen de los portales web. Bien es verdad que el sistema será capaz de permitir la creación de CVS a mano, pero la idea principal es la de obtener una gran cantidad de CVS de los portales web e introducirlos de golpe en el sistema, todo ello de manera automática. El objetivo de esto es el tiempo que se ahorra a la hora de tener que guardar la información. El crear los CVS a mano rellenando un formulario a poder insertar un archivo con un botón y darle a enviar, resulta bastante efectivo y es la esencia de todo el sistema, que las personas de selección de las empresas puedan introducir al sistema varios CVS a la vez para poder tratarlos posteriormente. Sin embargo, la opción de introducir CVS manualmente siempre es recomendable por si acaso se produce algún error a la hora de subirlo automáticamente, o no se desea guardar toda la información y solo quieren algo específico o simplemente es que no se tiene el CV en el formato específico.

Por tanto, hay que ver cómo son esos CVS que se van a importar en el sistema. El apartado de selección de las empresas tiene acceso particular a los portales web. Este acceso permite poder acceder a determinados CVS, ya que no se pueden acceder a ellos de manera pública.



The screenshot shows the InfoJobs search interface. At the top, there's a navigation bar with 'InfoJobs Let's work!', 'Consejos y noticias', 'Zona candidatos', 'consulta', and 'Cerrar sesión'. Below this, a breadcrumb trail reads 'Estás en: InfoJobs Empleo » Menú privado » CV encontrados'. The main search area contains several input fields and checkboxes:

- Palabras clave incluidas en el CV:** A text box containing 'J2EE, UML, RSA, HTML, XML, XSL' with a link 'busca en cualquier apartado del CV ayuda sobre cómo usar palabras clave'.
- Provincias de residencia:** A dropdown menu showing 'Madrid'.
- Empresa:** A dropdown menu set to 'coincide con' and an empty text box, with a link 'Buscar varias empresas'.
- Puesto:** A text box containing 'programad*, desarrollado*' with a link 'Buscar varios puestos'.
- Two checkboxes: 'Buscar sólo en la empresa actual' and 'Buscar sólo en el puesto actual', both currently unchecked.
- Conocimientos:** A text box with a '+ Añadir' button and a link 'Introduce conocimientos uno por uno ¿Cómo funcionan?'.
- A link 'Más criterios de búsqueda (idiomas, estudios, disponibilidad...)'.
- A blue button labeled 'Buscar CVs'.
- A link 'Buscar usando mis búsquedas guardadas »'.

Below the search area, it states '11 CV encontrados'.

Ilustración 9: Obtención de CVS -1

Esta pantalla se corresponde con la búsqueda por determinados criterios, como palabras clave, provincia, empresa...etc. En este caso, se puede ver como se han rellenado los campos de palabras clave incluidas en el CV, la provincia de residencia y el puesto (en una experiencia profesional).

El resultado son 11 CVS encontrados.

11 CV encontrados

¿Sueles buscar este tipo de perfiles?
[Guardar esta búsqueda](#)

Filtros

País
☒ España (16)

Provincia
☒ Madrid (11)
 Añade otras provincias
☐ Ciudad Real (1)
☐ Illes Balears (1)
☐ Salamanca (1)
☐ Valladolid (1)
☐ Asturias (1)

Población
☐ Madrid (6)
☐ Las Rozas de Madrid (2)
☐ Coslada (1)
☐ Mejorada del Campo (1)
☐ Valdemoro (1)

Edad
☐ de 25 a 35 (8)
☐ de 36 a 45 (3)

Categoría

Todos | Sólo con datos de contacto visibles | Sólo inscritos en mis ofertas | Mis favoritos

CVs seleccionados:

<input type="checkbox"/>	Nombre	Últ. conexión	Edad	Título	Población	Último puesto
<input type="checkbox"/>	Dur	03/05/16		I. Sup.		Arquitecto de software
<input type="checkbox"/>	Pala	27/02/16		Lic.		Project Manager
<input type="checkbox"/>	SANK	26/11/15		I. Téc.		Responsable Desarrollo proyecto Evolucion Mapfre
<input type="checkbox"/>	Candidato anónimo	26/08/15	43	Otr. cur.	Las rozas	Analista
<input type="checkbox"/>	Halc	18/05/15		I. Sup.		Sibel CRM - Oracle BI/ BIAPPS Senior Consultant
<input type="checkbox"/>	Vil	11/05/15		F.P. II		Senior Consultant
<input type="checkbox"/>	Hid	22/04/15		* I. Téc.		Analista Programador
<input type="checkbox"/>	Vak	06/04/15		Mas.		Analista programador
<input type="checkbox"/>	Ri	07/03/15		Mas.		Analista Programador J2EE
<input type="checkbox"/>	Palar	11/12/14		I. Téc.		Arquitecto Técnico

Ilustración 10: Obtención CVS -2

Esta es la lista de resultados que aparecen de la búsqueda. Por la protección de datos, no se muestran todos los datos de las 11 personas. Sin embargo, uno de los resultados es un "Candidato anónimo", y podemos acceder a él.

InfoJobs *Let's work!* Consejos y noticias Zona candidatos consulta Cerrar sesión

Estás en: [Página principal](#) > [Menú privado](#) > [CV encontrados](#) > CV

[« Volver a CVs encontrados](#) [» CV anterior](#) 4 de 11 [CV siguiente »](#)

CV de nuestra base de datos [Vista resumida](#) [Vista clásica](#)
 Los CVs se abren siempre en vista clásica [Cambiar](#)

CV completo (encontrado en InfoJobs)

Tienes Talent Search Sky activo. Puedes acceder a los datos de contacto de todos los CVs genéricos.

Actualización CV: 08/06/2015 [Compartir interés](#)

[★ Guardar en tus CV favoritos](#) [Imprimir](#)

[1 recomendación](#)

Datos personales

FOTO OPCIONAL

Nombre completo:	Prefero ocultarlo	E-mail:	
Sexo:	Hombre		
Fecha de nacimiento:	11/12/1972		
Código postal:	28290		
Población:	Las rozas		
Provincia:	Madrid		
NIF:	Prefero ocultarlo		
Teléfono móvil:	Prefero ocultarlo		
Mi página:	Prefero ocultarlo		

Ilustración 11: Obtención de CVS -3

Una vez accedido al CV, el personal de selección tiene la opción de poder guardarlo.

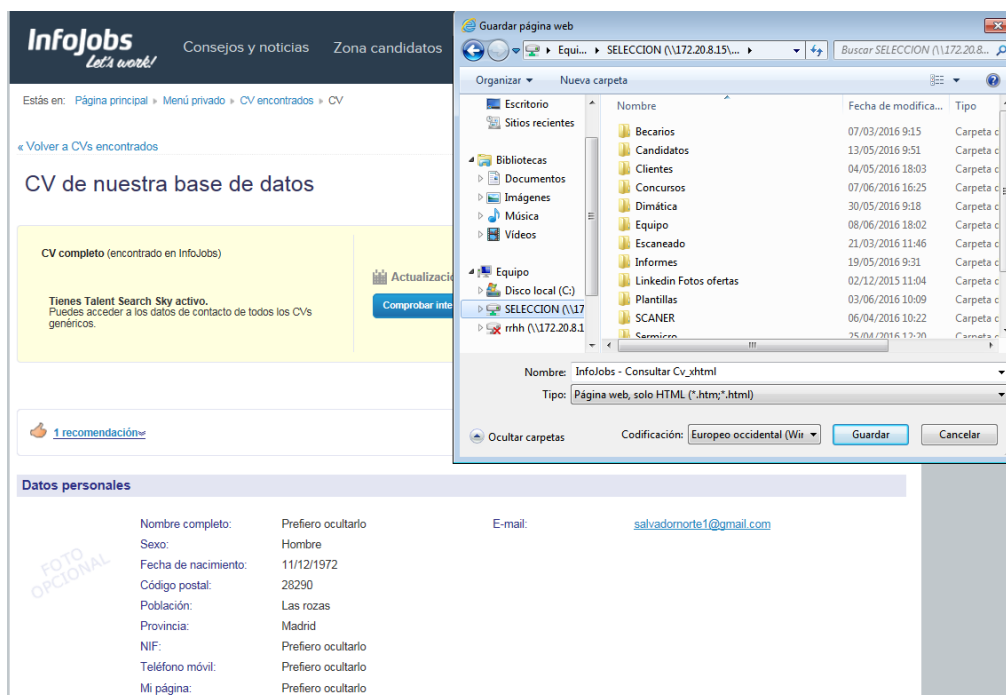


Ilustración 12: Obtención de CVS -4

El formato de guardado será en html. Todos los CVS que se pueden pasar al sistema deberán tener este formato y el sistema se encargará de leerlo y de guardar la información necesaria en la base de datos. De la misma manera se haría para CVS de Tecnoempleo.

4.2. Base de datos

Una vez analizado la parte más importante del sistema, vamos a pasar a describir la base de datos, elemento también fundamental ya que será donde se guarde toda la información de los CVS.

En primer lugar, hay que explicar el conector que se ha utilizado para unir la base de datos con el servidor. Se ha utilizado ODBC, que es un estándar de acceso a base de datos, que viene instalado por defecto en los equipos (en caso contrario se podría descargar de la página oficial de MySQL). Para poder utilizarlo, hay que realizar una configuración específica.

En primer lugar, hay que acceder a él, mediante:

C:\Windows\SysWOW64\odbcad32.exe

Al pinchar, saldrá la siguiente ventana:

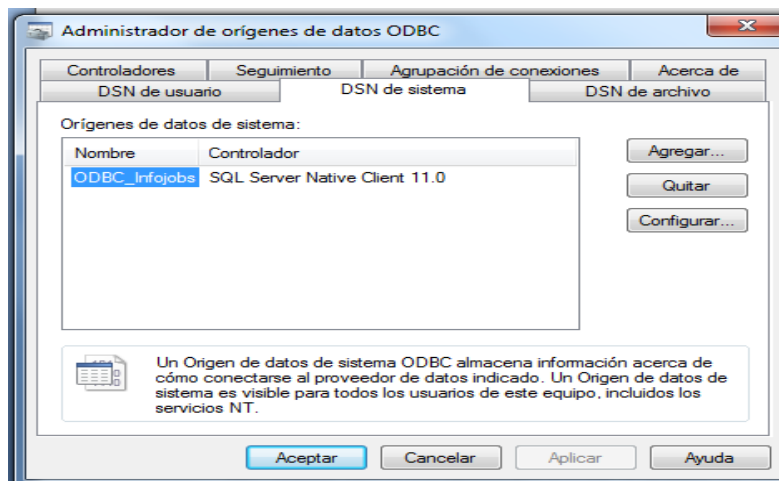


Ilustración 13: ODBC-1

Inicialmente aparecería vacío, pero habría que darle a la pestaña de agregar. Saldría lo siguiente:

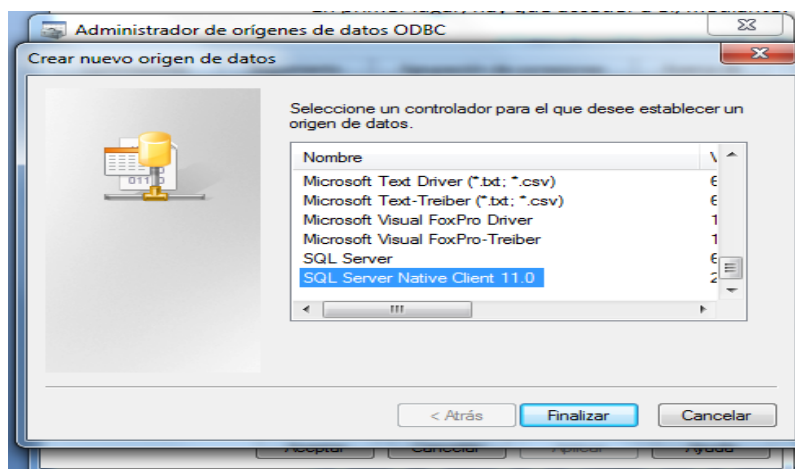


Ilustración 14: ODBC-2

Habría que seleccionar esta opción, que la que se corresponde con el SQL Server actualizado. Después, saldría lo siguiente:

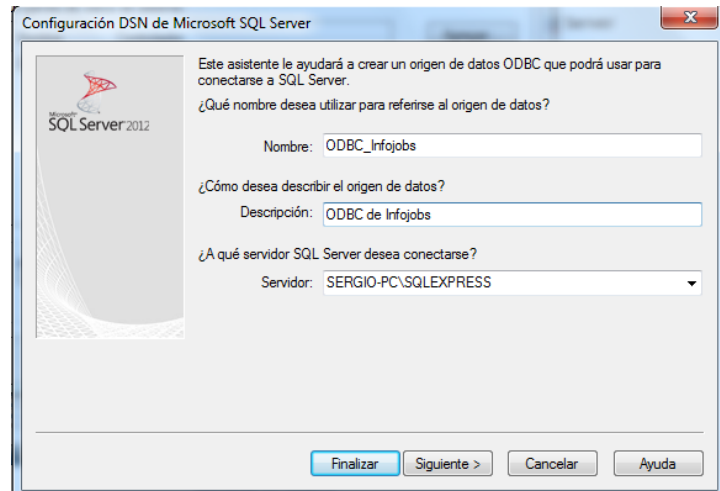


Ilustración 15: ODBC-3

Al rellenar esto y dar a siguiente, el conector detectará el servidor:

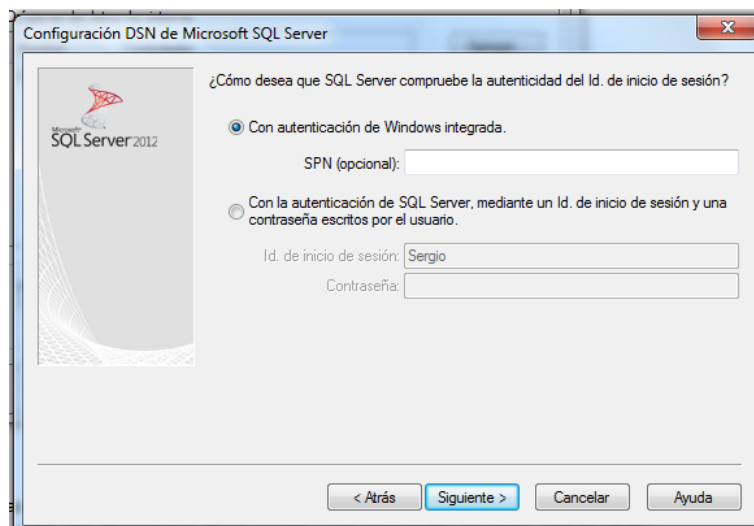


Ilustración 16: ODBC-4

En caso de tener usuario y contraseña se introducen, y si no se autentica con Windows. En este caso autenticamos con Windows ya que no he decidido crear una contraseña. Una vez dado en siguiente, el conector realiza una primera conexión con el SQL si todo ha ido bien.

Se mostraría la siguiente pantalla:

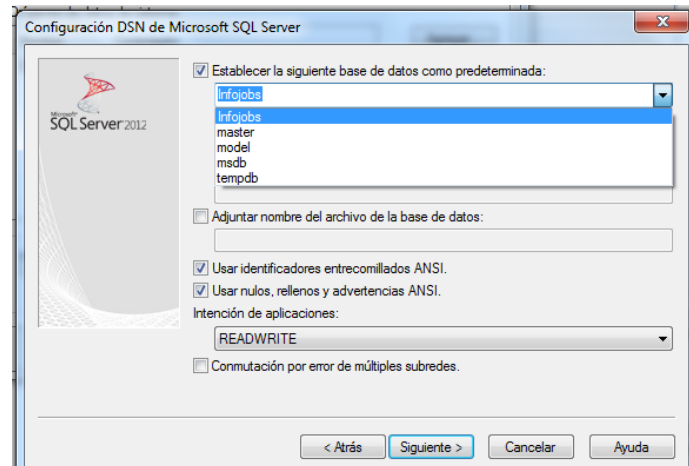


Ilustración 17: ODBC-5

Si todo ha ido correctamente, en el desplegable se mostrarán las bases de datos creadas en ese servidor, y vemos como todo ha ido correctamente al tener la base de datos "Infojobs". La seleccionamos y pinchamos en siguiente:

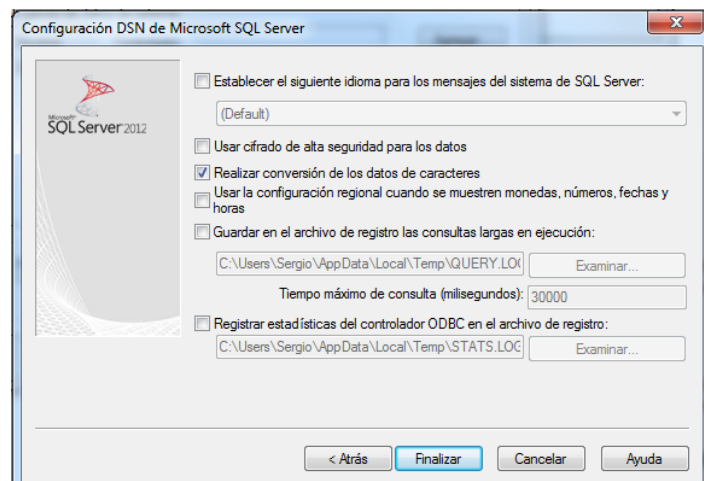


Ilustración 18: ODBC-6

En esta pantalla no habría que hacer nada, simplemente pinchar en finalizar. Por tanto, lo que hemos hecho aquí ha sido crear una conexión entre un servidor con una base de datos específica, de tal manera que podremos acceder a la base de datos como queramos. El lenguaje PHP se complementa muy bien con este conector, ya que dispone de una librería concreta para esta conexión, por lo que podremos utilizar PHP para poder hacer todos los accesos a la base de datos.

La base de datos se ha creado con la herramienta "Microsoft SQL Server 2012" mencionada antes en el apartado de medios software. Esta herramienta es muy sencilla de utilizar y basta con realizar dos o tres pasos para crear una base de datos.

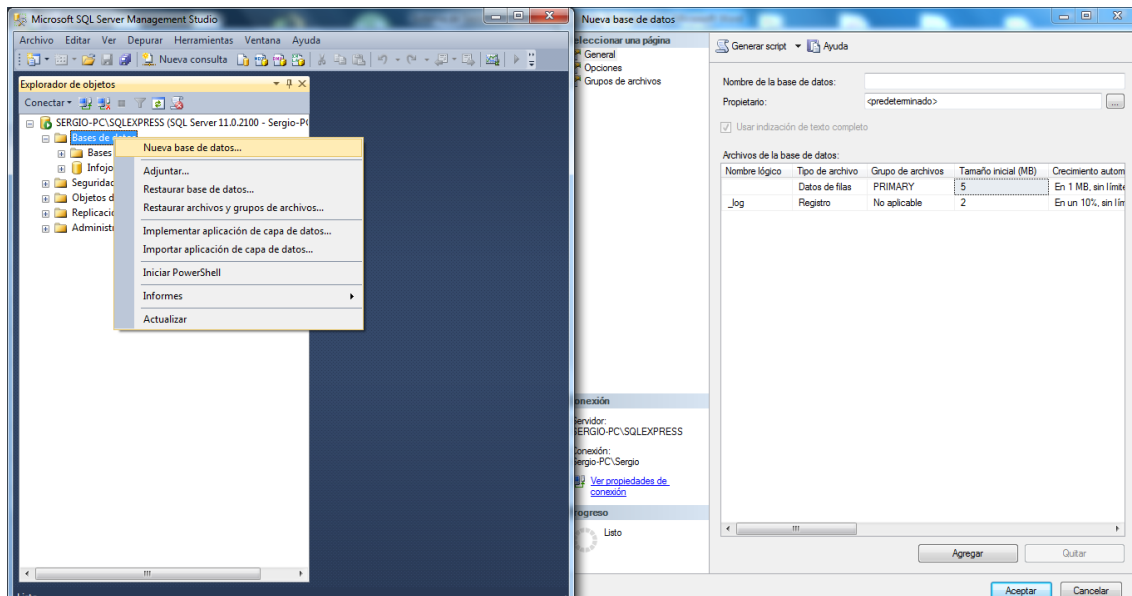


Ilustración 19: Ejemplo creación Base de datos

Una vez creada la base de datos, se pueden crear dentro tantas tablas como necesitemos, con sus relaciones entre sí, los atributos de las tablas y sus tipos...etc.

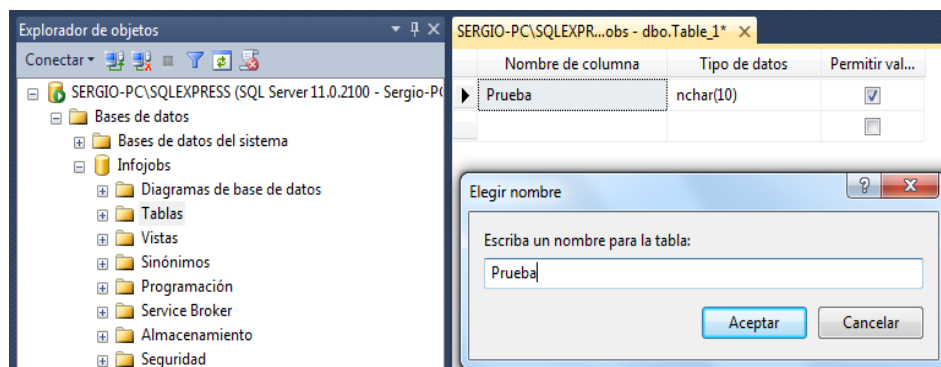


Ilustración 20: Ejemplo creación tabla

Cabe mencionar que es necesario un servidor al cual conectar el SQL, para poder trabajar. Este servidor es el mencionado antes también en el apartado de medios software (que crea el XAMPP). Una vez creada la base de datos, simplemente hay que hacer las inserciones correspondientes con los datos obtenidos de la lectura del CV.

4.3. Programa de recogida de información

A continuación, vamos a analizar lo que es el programa fundamental, que se encarga de leer el CV con formato html. El programa ha sido realizado con el lenguaje de programación PHP, y acepta CVS del portal web Infojobs y Tecnoempleo. Esta elección se debe a que son dos de los portales web más utilizados respecto al tema de CVS.

```
<?php
$myFile=$_FILES['fichero_usuario'];
$fileCount=count($myFile['name']);

for ($i = 0; $i < $fileCount; $i++) {
    $file=$_FILES['fichero_usuario'][$i]['tmp_name'];
    if (is_uploaded_file($file)) {
        comprobarFormato($file);
    }
    else {
        echo "Error:".$_FILES['fichero_usuario'][$i]['error'] . " ";
    }
}
}
```

Ilustración 21: Lectura de CVS para guardar

Esta captura muestra como se crea un contador para ver cuántos CVS a la vez se suben, y con un for se recorren todos esos CVS guardándolos en variables y llamando a la función comprobarFormato, que detectará que tipo de CV es, si de Infojobs o Tecnoempleo.

El programa funciona de la siguiente manera: recibirá un determinado número de CVS, y para cada uno, realizará un proceso de carga (is_uploaded_file, que es una función de PHP que comprueba que se ha cargado el archivo correctamente), que si tiene éxito, pasará a llamar a la función comprobarFormato. Esta función detectará que tipo de CV es, y pasará a llamar a la función insertarInfojobs o insertarTecnoempleo, según convenga.

```
<?php
$myFile=$_FILES['fichero_usuario'];
$fileCount=count($myFile['name']);

for ($i = 0; $i < $fileCount; $i++) {
    $file=$_FILES['fichero_usuario'][$i]['tmp_name'];
    if (is_uploaded_file($file)) {
        comprobarFormato($file);
    }
    else {
        echo "Error:".$_FILES['fichero_usuario'][$i]['error'] . " .";
    }
}

function comprobarFormato($file){
    $html = new simple_html_dom();
    $html = file_get_html($file);

    if(!is_null($variableComprobar = $html->find('title',0)->plaintext)){
        $comprobacion = strpos($variableComprobar, 'Tecnempleo');
        if($comprobacion !== false){
            insertarTecnempleo($file);
        }
        else{
            insertarInfojobs($file);
        }
    }
}

function insertarInfojobs($file){

function insertarTecnempleo($file){
```

Ilustración 22: Estructura del programa

Esta captura muestra de forma clara como es el proceso del programa. Al principio recibirá un CV, si se carga con éxito se ejecutará la función comprobarFormato, y dependiendo del tipo de CV se ejecutará una función de insertar u otra. En caso de que la carga no fuera exitosa, se mostraría un mensaje de error. Más abajo, en el apartado de desarrollo, se hará más hincapié en el funcionamiento de las funciones.

4.4. Página web

A continuación, se va a analizar donde se va a visualizar el sistema para poder interactuar con él. Se ha creado una página web, utilizando el lenguaje de programación JavaScript, el lenguaje de etiquetas html y todo ello acompañado de hojas de estilo css para la maquetación.

```
<div class="botones-cabecera">
    <button class="atras" id='cerrar' type="button" style='float:right' onclick="cerrarSesion()">Cerrar sesión
    <i class="glyphicon glyphicon-log-out"></i></button>
</div>
```

Ilustración 23: Ejemplo de html

```
function cerrarSesion(){
    location.href='/InfojobsPRE/CerrarSesion.php';
}
```

Ilustración 24: Ejemplo de javascript

```
341 .atras {
342     float: right !important;
343 }
```

Ilustración 25: Ejemplo de CSS

```
<?php
session_start();
if(isset($_SESSION)){
    session_destroy();
}
session_write_close();
header("Location:/InfojobsPRE/Inicio.php");
?>
```

Ilustración 26: Ejemplo de PHP

Estas capturas son un ejemplo de cada tipo. En primer lugar, un ejemplo de html, donde se crea un botón, cuya función al pinchar en él es cerrarSesion() (con la palabra clave onclick). Después tenemos la captura de javascript, que define la función cerrarSesion(), y se encargará de redirigir a una página concreta, en este caso a CerrarSesion.php. Después tenemos una propiedad CSS, que indica que para la clase "atrás", se le tiene que aplicar como importante que se coloquen a la derecha. En este caso, el botón tiene la clase "atrás" (por la palabra clave 'class'), por lo que saldrá colocado a la derecha de donde esté contenido (div, table...). Finalmente tenemos un código PHP, que se trata del archivo CerrarSesion.php. Este se encarga de destruir la sesión abierta por el usuario y de redirigir a la página de Index.php, que es la pantalla inicial para poder acceder a la aplicación.

Se ha creado un apartado de alta de CVS, donde hay disponible un botón para la subida de CVS en formato html, y también un formulario para crear un CV nuevo desde cero. Además, hay otro apartado de búsqueda, donde se introducen unos criterios y el sistema responde con los CVS que cumplen esos criterios. Se puede seleccionar un CV y visualizarlo, además de editarlo y exportarlo a un formato. Por otro lado, tenemos el último apartado correspondiente a peticiones. Se podrán dar de alta peticiones, donde se especificarán los requisitos de cada una, y después se podrá hacer una búsqueda de las peticiones. En cada petición, se podrán agregar CVS que el usuario que utilice el sistema considere oportuno, normalmente porque el CV cumple los requisitos de la petición, y podrá añadir uno o varios seguimientos sobre este CV, esto es, poder llevar un control sobre qué acciones se han realizado sobre un CV.

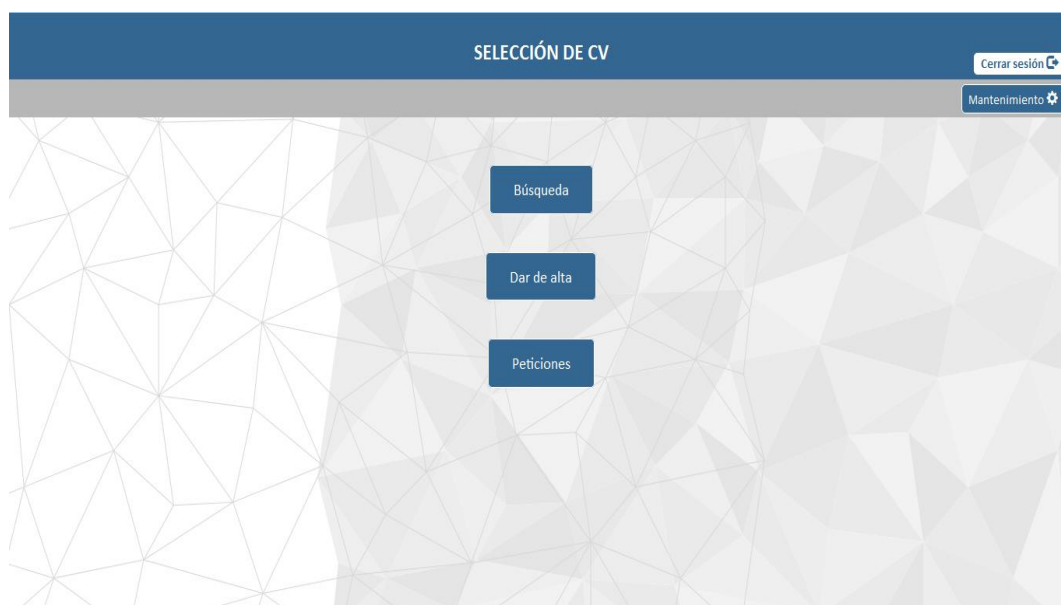


Ilustración 27: Ejemplo de página web de la aplicación

Se ha explicado con qué archivos se va a trabajar, como se van a tratar y donde se van a guardar los datos, y como todo ello se va a poder visualizar en una página web donde se podrán hacer todas las gestiones oportunas. Una vez hecho este análisis, pasaremos a explicar los requisitos del sistema y los casos de uso.

4.5. Requisitos del sistema

En este apartado se van a explicar detalladamente los requisitos para el SGCV propuesto. Estos requisitos vienen diferenciados en dos grupos para facilitar la comprensión de las exigencias del sistema: requisitos funcionales y no funcionales.

1- **Requisitos funcionales:** aquellos que describen la actuación y función del sistema. A su vez, se dividen en varios grupos:

- Requisitos de funcionamiento: determinan el funcionamiento del sistema: flujo de tareas, funcionalidad, etc.
- Requisitos de interfaz: determinan como será mostrado el sistema al usuario, y de qué manera se podrá interactuar con él.
- Requisitos de base de datos: determinan como se van a almacenar los datos en el sistema.

2- **Requisitos no funcionales:** aquellos que describen tareas no relacionadas con la funcionalidad, si no tareas como mantenimiento, supervisión, calidad, seguridad, etc.

La definición de ambos tipos de requisitos vendrá dada por la siguiente tabla:

ID		Título	
Tipo		Fecha	
Descripción			

Tabla 2: Tabla tipo de requisitos

Los campos de la tabla son:

- ID: identificador único que tendrá cada requisito. [TIPO(funcional F/no funcional NF)] [Subtipo(funcionamiento F/interfaz I/base de datos B)] [NUMERO]. Por ejemplo, FF01 correspondería con requisito funcional de funcionamiento, número 1, o NF05 correspondería con requisito no funcional número 5.
- Título: nombre de cada requisito.
- Fecha: hace referencia a la fecha de creación del requisito.
- Descripción: explicación concisa del requisito.

Requisitos funcionales

Requisitos de funcionamiento

ID	FF01	Título	Registro
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema debe permitir el registro de usuarios por parte de un "Administrador".		

Tabla 3: Requisito funcional de funcionamiento FF01

ID	FF02	Título	Registro de perfiles
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema debe permitir el registro de usuarios por parte de un "Administrador" con distintos perfiles de usuarios.		

Tabla 4: Requisito funcional de funcionamiento FF02

ID	FF03	Título	Iniciar sesión
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá al usuario la posibilidad de iniciar sesión en la aplicación.		

Tabla 5: Requisito funcional de funcionamiento FF03

ID	FF04	Título	Cerrar sesión
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá al usuario registrado la posibilidad de cerrar su sesión.		

Tabla 6: Requisito funcional de funcionamiento FF04

ID	FF05	Título	Búsqueda CV
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá a todo tipo de usuarios una búsqueda de CV.		

Tabla 7: Requisito funcional de funcionamiento FF05

ID	FF06	Título	Resultados búsqueda CV
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá a todo tipo de usuarios la información de cada CV de un resultado de búsqueda de CV.		

Tabla 8: Requisito funcional de funcionamiento FF06

ID	FF07	Título	Exportar CV
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo al usuario tipo "administrador" la posibilidad de exportar un CV a Word.		

Tabla 9: Requisito funcional de funcionamiento FF07

ID	FF08	Título	Editar CV
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo al usuario tipo "administrador" la posibilidad de editar la información de un CV.		

Tabla 10: Requisito funcional de funcionamiento FF08

ID	FF09	Título	Alta CV formulario
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo al usuario tipo "administrador" la posibilidad de dar de alta un candidato rellenando un formulario.		

Tabla 11: Requisito funcional de funcionamiento FF09

ID	FF10	Título	Alta CV automático
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo al usuario tipo "administrador" la posibilidad de dar de alta un candidato importando un CV de "Infojobs" y/o "Tecnoempleo".		

Tabla 12: Requisito funcional de funcionamiento FF10

ID	FF11	Título	Alta petición formulario
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo al usuario tipo "administrador" y "petición" la posibilidad de dar de alta una petición rellenando un formulario.		

Tabla 13: Requisito funcional de funcionamiento FF11

ID	FF12	Título	Búsqueda peticiones
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador", "gestor" y "petición" la posibilidad de buscar peticiones.		

Tabla 14: Requisito funcional de funcionamiento FF12

ID	FF13	Título	Resultado búsqueda peticiones
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador", "gestor" y "petición" la información de peticiones del resultado de una búsqueda de peticiones.		

Tabla 15: Requisito funcional de funcionamiento FF13

ID	FF14	Título	Editar peticiones
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" y "petición" la posibilidad de editar la información de una petición.		

Tabla 16: Requisito funcional de funcionamiento FF14

ID	FF15	Título	Gestionar peticiones
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" y "petición" la posibilidad de gestionar una petición.		

Tabla 17: Requisito funcional de funcionamiento FF15

ID	FF16	Título	Agregar CV a petición
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" y "petición" la posibilidad de agregar CV a una petición dentro de la gestión de la petición.		

Tabla 18: Requisito funcional de funcionamiento FF16

ID	FF17	Título	Añadir seguimiento a CV asociado a petición
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" y "petición" la posibilidad de añadir un seguimiento a un CV asociado a una petición dentro de la gestión de la petición.		

Tabla 19: Requisito funcional de funcionamiento FF17

ID	FF18	Título	Editar gestión de petición
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" y "petición" la posibilidad de editar la información contenida en la gestión de una petición.		

Tabla 20: Requisito funcional de funcionamiento FF18

ID	FF19	Título	Mantenimiento
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" realizar el mantenimiento de la aplicación.		

Tabla 21: Requisito funcional de funcionamiento FF19

ID	FF20	Título	Mantenimiento usuarios
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar usuarios.		

Tabla 22: Requisito funcional de funcionamiento FF20

ID	FF21	Título	Mantenimiento idiomas
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar idiomas.		

Tabla 23: Requisito funcional de funcionamiento FF21

ID	FF22	Título	Mantenimiento provincias
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar provincias.		

Tabla 24: Requisito funcional de funcionamiento FF22

ID	FF23	Título	Mantenimiento conocimientos
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar conocimientos.		

Tabla 25: Requisito funcional de funcionamiento FF23

ID	FF24	Título	Mantenimiento titulaciones
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar titulaciones.		

Tabla 26: Requisito funcional de funcionamiento FF24

ID	FF25	Título	Mantenimiento perfiles
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá sólo a usuarios tipo "administrador" la posibilidad de crear, modificar o eliminar perfiles.		

Tabla 27: Requisito funcional de funcionamiento FF25

ID	FF26	Título	Volver atrás
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá a todos los usuarios la posibilidad de volver a la pantalla anterior pinchando en el botón "Volver atrás".		

Tabla 28: Requisito funcional de funcionamiento FF26

ID	FF27	Título	Volver a inicio
Tipo	Funcionamiento	Fecha	17/06/2016
Descripción	El sistema ofrecerá a todos los usuarios la posibilidad de volver a la pantalla de inicio pinchando en el botón "Inicio".		

Tabla 29: Requisito funcional de funcionamiento FF27

Requisitos de interfaz

ID	FI01	Título	Identificación
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar y el botón "Iniciar sesión" para iniciar sesión.		

Tabla 30: Requisito funcional de interfaz FI01

ID	FI02	Título	Pantalla de inicio
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los distintos módulos que son accesibles (Búsqueda CV, alta candidato, peticiones, mantenimiento).		

Tabla 31: Requisito funcional de interfaz FI02

ID	FI03	Título	Filtros búsqueda CVS
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar y el botón de "Buscar" para realizar una búsqueda de CVS.		

Tabla 32: Requisito funcional de interfaz FI03

ID	FI04	Título	Mostrar resultados búsqueda CVS
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los resultados de la búsqueda de CVS, donde aparezca el nombre del propietario del CV.		

Tabla 33: Requisito funcional de interfaz FI04

ID	FI05	Título	Mostrar CV
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara la información que contiene un CV que ha sido accedido por una búsqueda de CVS, además de un botón "Editar" para poder modificar los datos (sólo administrador).		

Tabla 34: Requisito funcional de interfaz FI05

ID	FI06	Título	Editar CV
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos rellenos de la información del CV que podrán ser modificados, además de un botón "Borrar datos" que elimine el CV y un botón "Guardar cambios" que almacene las modificaciones añadidas.		

Tabla 35: Requisito funcional de interfaz FI06

ID	FI07	Título	Filtros búsqueda peticiones
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar y el botón de "Buscar" para realizar una búsqueda de peticiones.		

Tabla 36: Requisito funcional de interfaz FI07

ID	FI08	Título	Mostrar resultados búsqueda peticiones
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los resultados de la búsqueda de peticiones, donde aparezca el nombre, la fecha de creación y el estado de la petición.		

Tabla 37: Requisito funcional de interfaz FI08

ID	FI09	Título	Mostrar petición
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara la información que contiene una petición que ha sido accedido por una búsqueda de peticiones, además de un botón "Editar" para poder modificar los datos (sólo "administrador" y "gestor").		

Tabla 38: Requisito funcional de interfaz FI09

ID	FI10	Título	Gestionar petición
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los CV asociados a una petición, en un apartado distinto de los seguimientos de cada CV asociado. Además, un botón "Guardar cambios" para modificar la gestión de la petición, y un botón "Agregar CV" para poder añadir CVS a la petición.		

Tabla 39: Requisito funcional de interfaz FI10

ID	FI11	Título	Formulario alta candidato
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar y el botón "Guardar cambios" para poder dar de alta un candidato.		

Tabla 40: Requisito funcional de interfaz FI11

ID	FI12	Título	Formulario alta peticiones
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar y el botón "Guardar cambios" para poder dar de alta una petición.		

Tabla 41: Requisito funcional de interfaz FI12

ID	FI13	Título	Alta CV automático
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara un botón para seleccionar CV ("Seleccionar archivos") y otro botón para enviarlos ("Enviar ficheros").		

Tabla 42: Requisito funcional de interfaz FI13

ID	FI14	Título	Mantenimiento
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara un botón para cada sección de mantenimiento (usuarios, idiomas, provincias, conocimientos, titulaciones, perfiles).		

Tabla 43: Requisito funcional de interfaz FI14

ID	FI15	Título	Mantenimiento secciones - 1
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar un botón de "Añadir", "Modificar" y "Quitar" en cada sección de mantenimiento, que permita el tratamiento total de esa información.		

Tabla 44: Requisito funcional de interfaz FI15

ID	FI16	Título	Mantenimiento secciones - 2
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar de forma clara los campos a rellenar tanto para añadir como para modificar información, además de un botón de "Guardar" que añada nueva información o la modifique.		

Tabla 45: Requisito funcional de interfaz FI16

ID	FI17	Título	Botón cerrar sesión
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar arriba a la derecha en todas las pantallas un botón "Cerrar sesión" que permita cerrar la sesión al usuario.		

Tabla 46: Requisito funcional de interfaz FI17

ID	FI18	Título	Botón volver atrás
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar arriba a la derecha en todas las pantallas excepto en la de inicio un botón "Volver atrás" que permita retroceder a la pantalla anterior al usuario.		

Tabla 47: Requisito funcional de interfaz FI18

ID	FI19	Título	Botón Inicio
Tipo	Interfaz	Fecha	17/06/2016
Descripción	Se deberá mostrar arriba a la derecha en todas las pantallas excepto en la de inicio un botón "Inicio" que permita ir a la pantalla de inicio al usuario.		

Tabla 48: Requisito funcional de interfaz FI19

Requisitos de base de datos

ID	FB01	Título	Comprobar inicio de sesión
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá comprobar que el usuario y contraseña introducidos existen en la base de datos y son correctos para permitir el acceso a la aplicación del usuario que introduce los datos.		

Tabla 49: Requisito funcional de base de datos FB01

ID	FB02	Título	Crear mantenimiento
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá guardar en la base de datos la información que se añada o modifique en mantenimiento. Esta información corresponde con los usuarios del sistema, las provincias, los idiomas, los conocimientos, las titulaciones y los perfiles.		

Tabla 50: Requisito funcional de base de datos FB02

ID	FB03	Título	Distinción de mantenimiento
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá asignar un identificador único a cada usuario, provincia, idioma, conocimiento, titulación y perfil, que no podrá repetirse, a la hora de almacenar la información en la base de datos.		

Tabla 51: Requisito funcional de base de datos FB03

ID	FB04	Título	Crear candidato.
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá guardar en la base de datos la información de cada candidato, ya sea incluido por formulado o importado directamente.		

Tabla 52: Requisito funcional de base de datos FB04

ID	FB05	Título	Distinción de candidatos
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá asignar un identificador único, que no podrá repetirse, a cada candidato a la hora de almacenarlo en la base de datos.		

Tabla 53: Requisito funcional de base de datos FB05

ID	FB06	Título	Búsqueda CVS
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de extraer la información de la base de datos que se especifica en los filtros de búsqueda de CVS para poder mostrarla al usuario.		

Tabla 54: Requisito funcional de base de datos FB06

ID	FB07	Título	Edición de CVS
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de extraer la información del CV seleccionado del resultado de la búsqueda de CVS de la base de datos para que el usuario pueda verla y modificarla.		

Tabla 55: Requisito funcional de base de datos FB07

ID	FB08	Título	Modificar cambios de CVS
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de acceder a la base de datos y modificar la información necesaria sobre el CV que se esté editando del resultado de la búsqueda de CVS.		

Tabla 56: Requisito funcional de base de datos FB08

ID	FB09	Título	Crear petición
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá guardar en la base de datos la información de cada petición que se haya incluido en el formulario de alta de peticiones.		

Tabla 57: Requisito funcional de base de datos FB09

ID	FB10	Título	Distinción de peticiones
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá asignar un identificador único, que no podrá repetirse, a cada petición a la hora de almacenarla en la base de datos.		

Tabla 58: Requisito funcional de base de datos FB10

ID	FB11	Título	Búsqueda de peticiones
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de extraer la información de la base de datos que se especifica en los filtros de búsqueda de peticiones para poder mostrarla al usuario.		

Tabla 59: Requisito funcional de base de datos FB11

ID	FB12	Título	Edición de peticiones
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de extraer la información de la petición seleccionada del resultado de la búsqueda de peticiones de la base de datos para que el usuario pueda verla y modificarla.		

Tabla 60: Requisito funcional de base de datos FB12

ID	FB13	Título	Modificar cambios de peticiones
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de acceder a la base de datos y modificar la información necesaria sobre la petición que se esté editando del resultado de la búsqueda de peticiones.		

Tabla 61: Requisito funcional de base de datos FB13

ID	FB14	Título	Gestión de peticiones
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de acceder a la base de datos para mostrar la petición del resultado de búsqueda de peticiones, así como los CV asociados y seguimientos que tenga esa petición.		

Tabla 62: Requisito funcional de base de datos FB14

ID	FB15	Título	Agregar CV
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de acceder a la base de datos para guardar los CV que están asociados a la petición del resultado de búsqueda de peticiones.		

Tabla 63: Requisito funcional de base de datos FB15

ID	FB16	Título	Seguimiento
Tipo	Base de datos	Fecha	18/06/2016
Descripción	El sistema deberá ser capaz de acceder a la base de datos para guardar los seguimientos de cada CV asociado de la petición del resultado de búsqueda de peticiones.		

Tabla 64: Requisito funcional de base de datos FB16

Requisitos no funcionales

ID	NF01	Título	Identificador de usuario
Tipo	No funcional	Fecha	18/06/2016
Descripción	El usuario deberá de introducir su correo electrónico y contraseña para poder acceder a la aplicación.		

Tabla 65: Requisito no funcional NF01

ID	NF02	Título	Comprobar email
Tipo	No funcional	Fecha	18/06/2016
Descripción	El sistema comprobará durante el registro que la dirección de correo electrónico introducida por el administrador es válida: formato del email es correcto.		

Tabla 66: Requisito no funcional NF02

ID	NF03	Título	Accesibilidad
Tipo	No funcional	Fecha	18/06/2016
Descripción	Se debe asegurar que el sistema sea accesible desde cualquier terminal con acceso a Internet.		

Tabla 67: Requisito no funcional NF03

ID	NF04	Título	Navegadores
Tipo	No funcional	Fecha	18/06/2016
Descripción	La aplicación web debe funcionar en los navegadores de uso mayoritario: <ul style="list-style-type: none"> • Mozilla firefox, versión más reciente. • Google chrome, versión más reciente • Internet explorer, versiones 8, 9 y 10. 		

Tabla 68: Requisito no funcional NF04

ID	NF05	Título	LOPD
Tipo	No funcional	Fecha	18/06/2016
Descripción	El sistema deberá cumplir la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.		

Tabla 69: Requisito no funcional NF05

4.6. Casos de uso

En este apartado se van a explicar detalladamente los casos de uso para el SGCV propuesto, que servirán para especificar el comportamiento del sistema, así como la secuencia de interacciones entre el sistema y el usuario. De la misma manera que para los requisitos, estos casos de uso se definirán con una tabla que tiene la siguiente estructura:

ID		Título		Fecha	
Actores					
Referencias					
Objetivo					
Pre-condición					
Post-condición					
Resumen					

Tabla 70: Tabla tipo de casos de uso

Los campos de la tabla son:

- ID: identificador único para cada caso de uso. [TIPO][NÚMERO]
- Título: nombre de cada caso de uso.

- Fecha: hace referencia a cuando se creó el caso de uso.
- Referencias: establece aquellos requisitos a los que hace referencia el caso de uso.
- Objetivo: define el fin del caso de uso.
- Pre-condición: define las condiciones previas para que suceda el caso de uso.
- Post-condición: define el estado posterior del sistema, tras ocurrir el caso de uso.
- Resumen: breve descripción del contenido de cada caso de uso.

Casos de uso

A continuación se muestran todos los casos de uso del sistema:

ID	CU01	Título	Registrar cuenta	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF01, FF02, FF20, FB02, FB03				
Objetivo	Dar de alta un usuario en el sistema.				
Pre-condición	-Se debe rellenar el formulario de registro. -Se debe incluir un correo electrónico válido. -No se puede repetir correo electrónico que ya exista en la base de datos.				
Post-condición	-El usuario queda registrado en la base de datos del sistema. -El usuario puede iniciar sesión en el sistema.				
Resumen	El administrador accede al módulo de mantenimiento, y selecciona "Añadir usuario". Rellena el formulario con los datos que el usuario le haya proporcionado y pasa a darle de alta. Posteriormente le comunica al usuario su contraseña para acceder con su correo electrónico.				

Tabla 71: Caso de uso CU01

ID	CU02	Título	Modificar cuenta	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF01, FF02, FF20, FI14, FI15, FI16, FB02, FB03				
Objetivo	Eliminar o modificar la cuenta de un usuario.				
Pre-condición	- Que haya usuarios registrados en el sistema.				
Post-condición	- Los datos de la cuenta elegida son modificados o eliminados.				
Resumen	El administrador accede al módulo de mantenimiento, y entra en la parte de usuarios. Selecciona un usuario en concreto, y modifica o elimina sus datos pinchando en "Guardar".				

Tabla 72: Caso de uso CU02

ID	CU03	Título	Mantenimiento	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF01, FF02, FF20, FF21, FF22, FF23, FF24, FF25, FI14, FI15, FI16, FB02, FB03				
Objetivo	Eliminar o modificar la información de mantenimiento.				
Pre-condición	-				
Post-condición	- Los datos elegidos por el administrador son modificados o eliminados.				
Resumen	El administrador accede al módulo de mantenimiento, y puede acceder a cualquier parte (usuarios, idiomas, provincias, conocimientos, titulaciones, perfiles). Selecciona un campo y modifica o elimina su información pinchando en "Guardar".				

Tabla 73: Caso de uso CU03

ID	CU04	Título	Iniciar sesión	Fecha	18/06/2016
Actores	Administrador, petición, gestor, usuario externo.				
Referencias	FF01, FF02, FF03, FI01, FB01, NF01, NF02, NF03				
Objetivo	Iniciar sesión en el sistema.				
Pre-condición	<ul style="list-style-type: none"> - El usuario debe encontrarse registrado en la base de datos del sistema. - Se debe rellenar correctamente el formulario de inicio de sesión. - El usuario no debe tener iniciada ya una sesión. 				
Post-condición	<ul style="list-style-type: none"> - La sesión del usuario queda iniciada. - El usuario accede a la pantalla de inicio de la aplicación. 				
Resumen	Cualquier usuario rellena el formulario de inicio de sesión introduciendo su correo electrónico y contraseña, y accede a la pantalla de inicio de la aplicación pinchando en "Iniciar sesión".				

Tabla 74: Caso de uso CU04

ID	CU05	Título	Cerrar sesión	Fecha	18/06/2016
Actores	Administrador, petición, gestor, usuario externo.				
Referencias	FF01, FF02, FF04, FI17				
Objetivo	Cerrar sesión en el sistema.				
Pre-condición	<ul style="list-style-type: none"> - El usuario debe encontrarse en la base de datos del sistema. - Debe existir una opción para cerrar sesión. - El usuario debe tener una sesión abierta. 				
Post-condición	<ul style="list-style-type: none"> - La sesión del usuario queda cerrada. - Se redirige automáticamente a la pantalla de iniciar sesión. 				
Resumen	Un usuario registrado e iniciado en la aplicación cierra su sesión pinchando en el botón "Cerrar sesión", y ésta se cierra.				

Tabla 75: Caso de uso CU05

ID	CU06	Título	Dar de alta manual	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF09, FI11, FB04, FB05				
Objetivo	Dar de alta un candidato.				
Pre-condición	<ul style="list-style-type: none"> - Se debe rellenar el formulario de alta de candidato. - Hay que introducir valores correctos en los campos (correo electrónico, DNI, etc). 				
Post-condición	- El usuario queda guardado en la base de datos.				
Resumen	El administrador accede al módulo de dar de alta, rellena el formulario con los datos correctos y pincha en el botón "Guardar cambios". La información queda registrada en la base de datos.				

Tabla 76: Caso de uso CU06

ID	CU07	Título	Dar de alta automática	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF10, FI13, FB04, FB05				
Objetivo	Dar de alta un candidato.				
Pre-condición	- Disponer del fichero en formato html del CV correspondiente.				
Post-condición	- El usuario queda guardado en la base de datos.				
Resumen	El administrador accede al módulo de dar de alta, y dentro accede a la parte de importar CV existente. Selecciona el CV concreto y pincha en "Enviar fichero(s)". La información del CV queda registrada en la base de datos.				

Tabla 77: Caso de uso CU07

ID	CU08	Título	Búsqueda de CV	Fecha	18/06/2016
Actores	Administrador, petición, gestor, usuario externo.				
Referencias	FF05, FI03, FI04, FB06				
Objetivo	Encontrar uno o varios CVS mediante una búsqueda.				
Pre-condición	- Que haya información de CVS guardada en la base de datos.				
Post-condición	- Se muestran los resultados de CVS que coincidan con los filtros de búsqueda.				
Resumen	Cualquier usuario accede al módulo de búsqueda, y rellena los filtros de búsqueda. El sistema responde con los CVS que contienen esa búsqueda al pinchar en "Buscar".				

Tabla 78: Caso de uso CU08

ID	CU09	Título	Ver CV	Fecha	18/06/2016
Actores	Administrador, petición, gestor, usuario externo.				
Referencias	FF05, FF06, FI05, FB06				
Objetivo	Ver la información de un CV.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de CVS guardada en la base de datos. - Que se haya realizado una búsqueda de CV con éxito. 				
Post-condición	- Se ve de forma clara toda la información contenida en el CV.				
Resumen	Cualquier usuario realiza una búsqueda de CV con determinados filtros, y selecciona uno de los resultados. A continuación verá toda la información del CV.				

Tabla 79: Caso de uso CU09

ID	CU10	Título	Exportar CV	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF05, FF06, FF07				
Objetivo	Exportar la información de un CV a Word.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de CVS guardada en la base de datos. - Que se haya realizado una búsqueda con éxito. - Que se esté visualizando la información de un CV. 				
Post-condición	- Se abre una ventana con opción a guardar el documento Word o simplemente abrirlo.				
Resumen	El usuario administrador que esté visualizando la información de un CV, podrá pinchar en "Exportar a Word" para generar un documento Word.				

Tabla 80: Caso de uso CU10

ID	CU11	Título	Editar CV	Fecha	18/06/2016
Actores	Administrador.				
Referencias	FF05, FF06, FF08, FI06, FB07, FB08				
Objetivo	Modificar la información de un CV.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de CVS guardada en la base de datos. - Que se haya realizado una búsqueda de CV con éxito. - Que se esté visualizando la información de un CV. 				
Post-condición	- Se modifica la información del CV en la base de datos.				
Resumen	El usuario administrador que esté visualizando la información de un CV, podrá pinchar en el botón "Editar datos". Dentro podrá modificar la información que guardará con el botón "Guardar cambios", o borrar el CV con el botón "Borrar datos".				

Tabla 81: Caso de uso CU11

ID	CU012	Título	Dar de alta peticiones	Fecha	18/06/2016
Actores	Administrador, petición, gestor.				
Referencias	FF11, FI12, FB09, FB10				
Objetivo	Dar de alta una petición.				
Pre-condición	- Se debe rellenar el formulario de alta de peticiones.				
Post-condición	- La petición queda guardada en la base de datos.				
Resumen	Cualquier usuario excepto el usuario externo, rellena el formulario de alta de peticiones y pincha en el botón "Guardar cambios". La petición queda guardada en la base de datos.				

Tabla 82: Caso de uso CU12

ID	CU13	Título	Búsqueda de petición	Fecha	18/06/2016
Actores	Administrador, petición, gestor.				
Referencias	FF12, FI07, FI08, FB11				
Objetivo	Ver la información de una petición.				
Pre-condición	- Que haya información de peticiones guardada en la base de datos.				
Post-condición	- Se muestran los resultados de peticiones que coincidan con los filtros de búsqueda.				
Resumen	Cualquier usuario excepto el usuario externo accede al módulo de peticiones, y dentro al de búsqueda de peticiones. Rellena los filtros de búsqueda y el sistema responde con las peticiones que contienen esa búsqueda al pinchar en "Buscar".				

Tabla 83: Caso de uso CU13

ID	CU14	Título	Ver petición	Fecha	18/06/2016
Actores	Administrador, petición, gestor.				
Referencias	FF12, FF13, FI07, FI08, FB11				
Objetivo	Ver la información de una petición.				
Pre-condición	- Que haya información de peticiones guardada en la base de datos. - Que se haya realizado una búsqueda de petición con éxito.				
Post-condición	- Se ve de forma clara toda la información contenida en la petición.				
Resumen	Cualquier usuario excepto el usuario externo realiza una búsqueda de petición con determinados filtros, y selecciona uno de los resultados. A continuación verá toda la información de la petición.				

Tabla 84: Caso de uso CU14

ID	CU15	Título	Editar petición	Fecha	18/06/2016
Actores	Administrador, petición.				
Referencias	FF12, FF13, FF14, FI09, FB12, FB13				
Objetivo	Modificar la información de una petición				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de peticiones guardada en la base de datos. - Que se haya realizado una búsqueda de peticiones con éxito. - Que se esté visualizando la información de una petición. 				
Post-condición	- Se modifica la información de la petición en la base de datos.				
Resumen	El usuario administrador o petición que esté visualizando la información de una petición, podrá pinchar en el botón "Editar datos". Dentro podrá modificar la información que guardará con el botón "Guardar cambios", o borrar la petición con el botón "Borrar datos".				

Tabla 85: Caso de uso CU15

ID	CU16	Título	Gestionar petición	Fecha	18/06/2016
Actores	Administrador, petición.				
Referencias	FF12, FF15, FI10, FB14, FB15, FB16				
Objetivo	Gestionar una petición.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de peticiones guardada en la base de datos. - Que se haya realizado una búsqueda de petición con éxito. 				
Post-condición	- Se accede al apartado de gestión de una petición.				
Resumen	Un usuario administrador o petición realizan una búsqueda de una petición rellenando los filtros de búsqueda de petición. El resultado ofrece un botón que indica "Gestión de la petición". Al pinchar se podrá acceder a la gestión de la petición.				

Tabla 86: Caso de uso CU16

ID	CU17	Título	Agregar CV a petición	Fecha	18/06/2016
Actores	Administrador, petición.				
Referencias	FF12, FF15, FF16, FF18, FI10, FB15, FB16				
Objetivo	Agregar un CV a una petición.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de peticiones guardada en la base de datos. - Que se haya realizado una búsqueda de petición con éxito. -Que se haya accedido a la gestión de la petición. 				
Post-condición	- Se agregan CVS asociados a la petición.				
Resumen	Un usuario administrador o petición acceden a la gestión de una petición del resultado de búsqueda de peticiones. En el apartado CVS asociados pinchan en "Agregar CV" y realizan la búsqueda de un CV que se quiera agregar pinchando en "Agregar a petición". Se guardarán los datos con el botón "Guardar cambios".				

Tabla 87: Caso de uso CU17

ID	CU18	Título	Crear seguimiento de CV en petición	Fecha	18/06/2016
Actores	Administrador, petición.				
Referencias	FF12, FF15, FF17, FF18, FI10				
Objetivo	Crear un seguimiento sobre un CV asociado a una petición.				
Pre-condición	<ul style="list-style-type: none"> - Que haya información de peticiones guardada en la base de datos. - Que se haya realizado una búsqueda de petición con éxito. -Que se haya accedido a la gestión de la petición. 				
Post-condición	- Se crea el seguimiento de ese CV asociado en esa petición.				
Resumen	Un usuario administrador o petición acceden a la gestión de una petición del resultado de búsqueda de peticiones. En el apartado de seguimiento, podrán crear seguimientos con el botón "Añadir seguimiento". Se guardarán los datos con el botón "Guardar cambios".				

Tabla 88: Caso de uso CU18

Diagramas de caso de uso

En este apartado se van a mostrar unos diagramas de casos de uso, que reflejarán de forma muy esquemática las posibles acciones de cada usuario. Estos diagramas se dividen en 4 debido a los tipos de usuarios que pueden utilizar el sistema.

Usuario "Administrador"

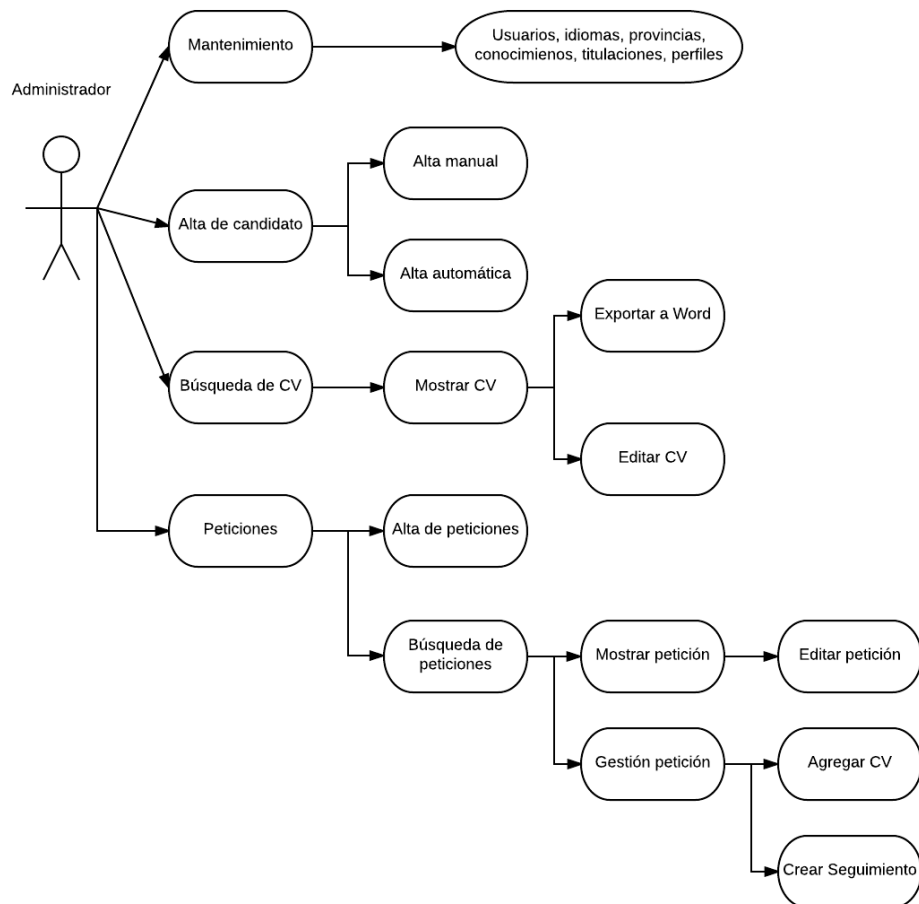


Ilustración 28: Usuario "Administrador"

El usuario "Administrador" puede realizar las siguientes actividades:

- Mantenimiento de la aplicación: crear usuarios, incluir idiomas, provincias, perfiles, etc. Además, puede editarlos o eliminarlos.
- Alta de candidato: el administrador será el único que pueda incluir candidatos en la aplicación, ya sea de forma manual o de manera automática importando un CV en formato html.
- Búsqueda de CV: buscar CVS por criterios de búsqueda, ver su información, poder exportarla a formato "Word" y poder modificar o borrar esa información.
- Peticiones: podrá crear peticiones rellenando un formulario, así como ver su información y editarla. Además, podrá gestionar la petición, agregando CVS o creando seguimientos sobre esos CVS asociados.

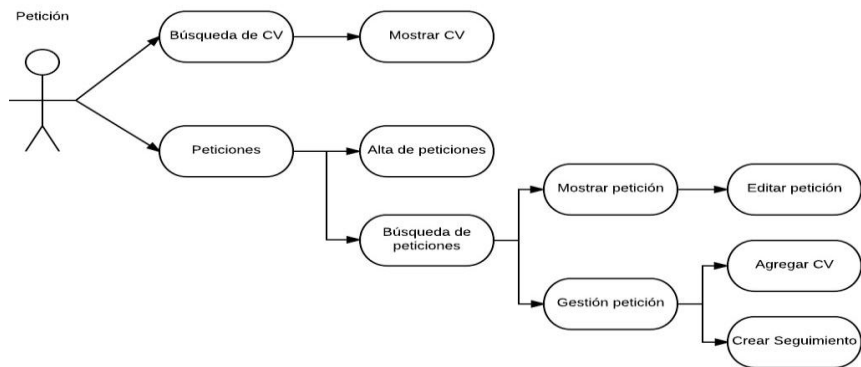
Usuario "Petición"

Ilustración 29: Usuario "Petición"

El usuario "Petición" puede realizar las siguientes actividades:

- Búsqueda de CV: buscar CVS por criterios de búsqueda y ver su información.
- Peticiones: podrá crear peticiones rellenando un formulario, así como ver su información y editarla. Además, podrá gestionar la petición, agregando CVS o creando seguimientos sobre esos CVS asociados.

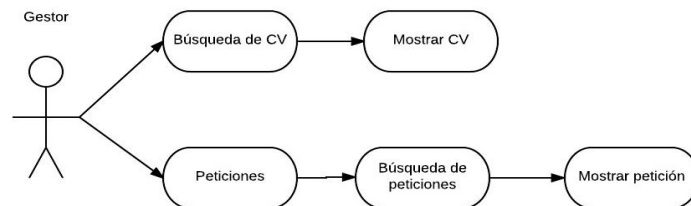
Usuario "Gestor"

Ilustración 30: Usuario "Gestor"

El usuario "Gestor" puede realizar las siguientes actividades:

- Búsqueda de CV: buscar CVS por criterios de búsqueda y ver su información.
- Peticiones: podrá buscar peticiones y ver su información.

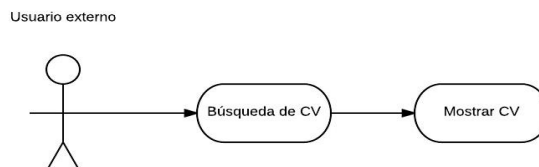
Usuario "Usuario externo"

Ilustración 31: Usuario "Usuario externo"

El usuario "Usuario externo" puede realizar las siguientes actividades:

- Búsqueda de CV: buscar CVS por criterios de búsqueda y ver su información.

5. Desarrollo

En este apartado se explicará con detalle todo el desarrollo realizado del SGCV. Hay tres partes diferenciadas en el desarrollo del sistema, estas son: base de datos, programa y página web.

5.1. Base de datos

En primer lugar, lo primero a desarrollar ha sido la base de datos donde se guarda toda la información de cada CV. Se ha realizado un estudio de qué información muestra un CV, analizando CVS que obteníamos de los portales web. Además, se han pensado nuevas ideas para incluir información adicional que pudiera ser útil. De esta manera, se creó un diseño relacional con las tablas, atributos y relaciones correspondientes. Una vez creado, el siguiente paso fue llevar este diseño al SQL Server. La base de datos queda de la siguiente manera:

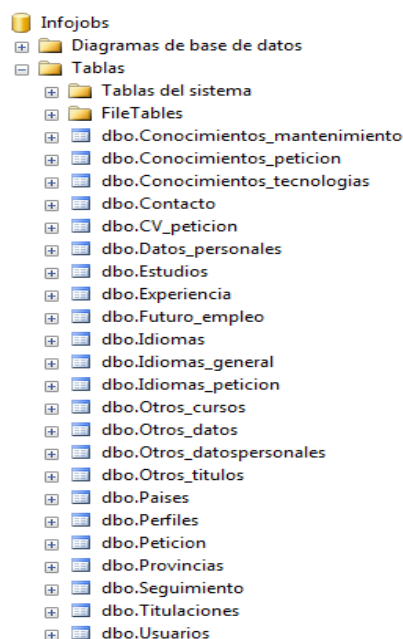


Ilustración 32: Base de datos

Como podemos ver, la base de datos se llama "Infojobs". Cabe mencionar que primero se creó una base de datos más pequeña, y que a medida se avanzaba en el desarrollo, se iban realizando modificaciones en función de nuevas necesidades. Además, todas las tablas tienen un identificador para cada tupla, el cual se genera con una función de PHP llamada NewGuid que genera automáticamente un guid. La función es la siguiente y está incluida en el fichero funciones.php:

```
function NewGuid() {
    $s = strtoupper(md5(uniqid(rand(), true)));
    $guidText =
        substr($s, 0, 8) . '-' .
        substr($s, 8, 4) . '-' .
        substr($s, 12, 4) . '-' .
        substr($s, 16, 4) . '-' .
        substr($s, 20);
    return $guidText;
}
```

Ilustración 33: Función NewGuid()

Finalmente, este ha sido el resultado y podemos ver todas las tablas que hay creadas. Estas tablas son las siguientes:

Conocimientos_mantenimiento

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
conocimiento	nvarchar(500)	<input type="checkbox"/>

Ilustración 34: Tabla Conocimientos_mantenimiento

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el conocimiento.

Esta tabla tiene la función de disponer de todos los conocimientos posibles. Según se vayan incorporando CVS, ya sea de forma automática o a mano, si se inserta un conocimiento que no existe en la base de datos, se insertará en esta tabla. Posteriormente, a la hora de querer incluir un conocimiento a mano, se dispondrá de un select que tendrá como valores los conocimientos incluidos en esta tabla. En este caso todos los campos son obligatorios, el identificador es de tipo guid y el conocimiento es un string con capacidad máxima de 500 caracteres.

Contacto

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
tipo_contacto	nvarchar(100)	<input type="checkbox"/>

Ilustración 35: Tabla Contacto

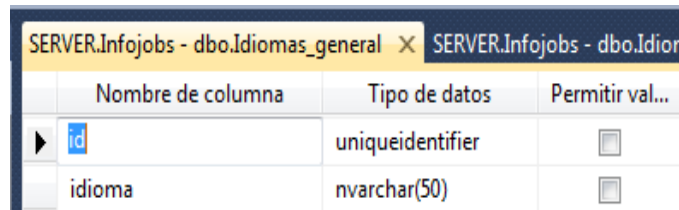
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el tipo de contacto de esa persona.

La función de esta tabla es igual que la de conocimientos_mantenimiento. Aquí se guardarán todos los contactos que se vayan insertando en caso de que no estuvieran antes. Los contactos son por ejemplo Facebook, Twitter...etc. Ambos campos son obligatorios, el id es de tipo guid y el tipo de contacto es un string de 100 caracteres máximo.

Idiomas_general

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
idioma	nvarchar(50)	<input type="checkbox"/>

Ilustración 36: Tabla Idiomas_general

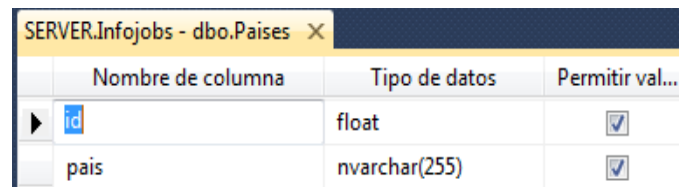
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece la categoría en concreto.
- El valor de la categoría o subcategoría de la experiencia profesional.
- Un identificador de la experiencia donde se aplican las categorías.

La función de esta tabla es igual que la de conocimientos_mantenimiento o Contacto. Aquí se guardarán todos los idiomas que se vayan insertando en caso de que no estuvieran antes. Los idiomas son por ejemplo Inglés, Francés, Italiano...etc. Ambos campos son obligatorios, el id es de tipo guid y el idioma es un string de 50 caracteres máximo.

Países

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	float	<input checked="" type="checkbox"/>
pais	nvarchar(255)	<input checked="" type="checkbox"/>

Ilustración 37: Tabla Países

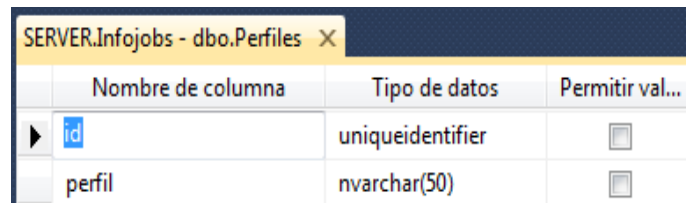
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir un país.

La función de esta tabla es la de guardar todos los países del mundo, para que a la hora de introducir un país, se haga mediante un select donde aparezcan todos los países, ya que es una cosa que existe como tal y no va a cambiar. No tiene sentido dejar al usuario introducir cualquier cosa, ya que los países son los que son. El identificador es de tipo float, ya que en este caso, como no se va a manipular esta tabla y solo se hará una carga inicial, los identificadores de los países van del 1 en adelante. El campo país es un string de 255 caracteres como máximo.

Perfiles

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
perfil	nvarchar(50)	<input type="checkbox"/>

Ilustración 38: Tabla Perfiles

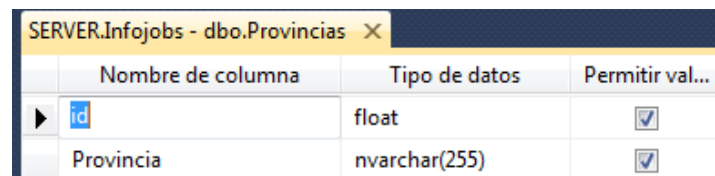
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el perfil.

La función de esta tabla es la de albergar los distintos perfiles de usuario a la hora de acceder y utilizar la aplicación. El identificador es de tipo guid y los dos campos son obligatorios. El perfil es un string de 50 caracteres como máximo.

Provincias

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	float	<input checked="" type="checkbox"/>
Provincia	nvarchar(255)	<input checked="" type="checkbox"/>

Ilustración 39: Tabla Provincias

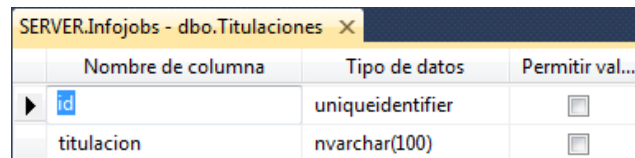
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir la provincia.

La función de esta tabla es la misma que la de países. Las provincias son las que son y aquí se almacenarán todas las provincias de España. El identificador es de tipo float, ya que en este caso, como no se va a manipular esta tabla y solo se hará una carga inicial, los identificadores de las provincias van del 1 en adelante. El campo provincia es un string de 255 caracteres como máximo.

Titulaciones

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
titulacion	nvarchar(100)	<input type="checkbox"/>

Ilustración 40: Tabla Titulaciones

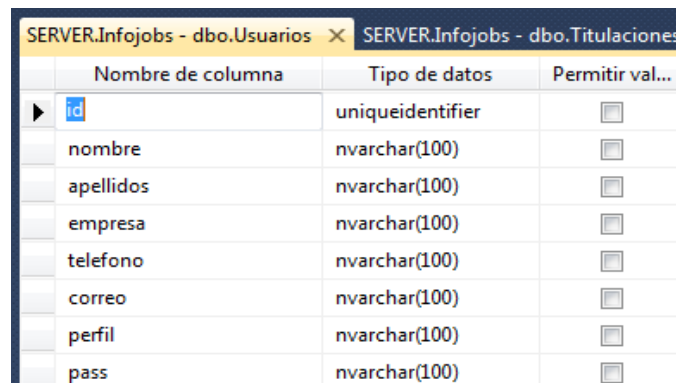
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para la titulación.

La función de esta tabla es la de guardar una serie de titulaciones predefinidas que no podrán cambiarse y que serán las opciones que tendrán los usuarios a la hora de elegir titulación. El identificador es de tipo guid y ambos campos son obligatorios. La titulación es un string de 100 caracteres como máximo.

Usuarios

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
nombre	nvarchar(100)	<input type="checkbox"/>
apellidos	nvarchar(100)	<input type="checkbox"/>
empresa	nvarchar(100)	<input type="checkbox"/>
telefono	nvarchar(100)	<input type="checkbox"/>
correo	nvarchar(100)	<input type="checkbox"/>
perfil	nvarchar(100)	<input type="checkbox"/>
pass	nvarchar(100)	<input type="checkbox"/>

Ilustración 41: Tabla Usuarios

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el nombre del usuario.
- Un campo para incluir los apellidos del usuario.
- Un campo para incluir la empresa del usuario.
- Un campo para incluir el teléfono del usuario.
- Un campo para incluir el correo del usuario.
- Un campo para incluir el perfil del usuario.
- Un campo para incluir la contraseña del usuario.

La función de esta tabla es la de guardar los datos sobre un usuario que accede a la aplicación y navega por ella. El identificador es de tipo guid y todos los campos son obligatorios.

Datos_personales

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
nombre_completo	nvarchar(100)	<input type="checkbox"/>
email	nvarchar(100)	<input type="checkbox"/>
sexo	nvarchar(10)	<input type="checkbox"/>
fecha_nacimiento	nvarchar(20)	<input type="checkbox"/>
direccion	nvarchar(100)	<input checked="" type="checkbox"/>
codigo_postal	nvarchar(10)	<input type="checkbox"/>
poblacion	nvarchar(50)	<input type="checkbox"/>
provincia	nvarchar(50)	<input type="checkbox"/>
NIF	nvarchar(9)	<input checked="" type="checkbox"/>
telefono_fijo	nvarchar(20)	<input checked="" type="checkbox"/>
telefono_movil	nvarchar(20)	<input checked="" type="checkbox"/>
telefono_extranjero	nvarchar(20)	<input checked="" type="checkbox"/>

Ilustración 42: Tabla Datos_personales

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el nombre completo de la persona.
- Un campo para incluir el email de la persona.
- Un campo para incluir el sexo de la persona.
- Un campo para incluir la fecha de nacimiento de la persona.
- Un campo para incluir la dirección de la persona.
- Un campo para incluir el código postal de la persona.
- Un campo para incluir la población de la persona.
- Un campo para incluir la provincia de la persona.
- Un campo para incluir el NIF de la persona.
- Un campo para incluir el teléfono fijo de la persona.
- Un campo para incluir el teléfono móvil de la persona.
- Un campo para incluir el teléfono extranjero de la persona.

Esta tabla es la principal de toda la base de datos. El identificador de esta tabla, es el identificador que se relacionará con las demás tablas en las que sea necesario incluir a una persona.

Así por ejemplo si una persona tiene Estudios, la relación sería la siguiente:

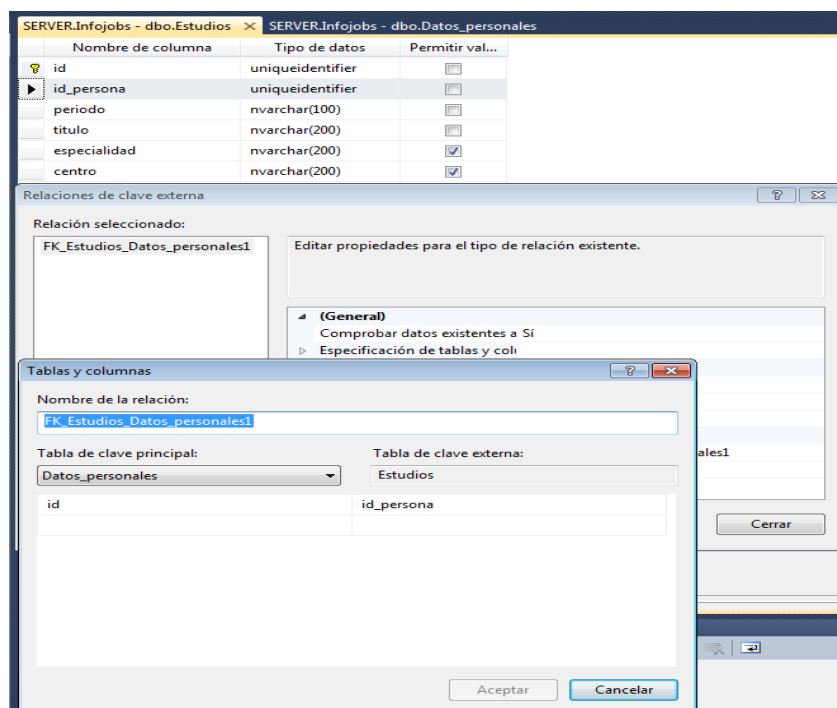


Ilustración 43: Ejemplo de relación entre tablas

En la tabla estudios, por cada tupla tendremos un estudio diferente. Este estudio tiene que corresponder con una persona que sea la que haya realizado ese estudio. En la tabla Estudio tenemos un identificador `id_persona`, que se corresponde con el `id` de la tabla datos personales. Por tanto, primero tenemos que tener insertados las personas en la tabla datos personales, y una vez insertados, a la hora de insertar cada estudio de cada persona, se relacionará al estudio con la persona mediante el `id_persona` que será clave ajena del `id` de la tabla datos personales.

La función de esta tabla es guardar los datos personales de cada persona, y a su vez servir de punto de partida a la hora de crear las demás tablas, ya que todas las personas que se usen en otras tablas (estudios, idiomas...) heredarán de esta tabla. El `id` es de tipo `guid`, y la dirección y los tres teléfonos no son obligatorios, lo demás sí. Exceptuando el `id`, los demás campos son de tipo `string` variando su capacidad máxima.

Otros_datospersonales

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
id_tipo_contacto	uniqueidentifier	<input type="checkbox"/>
contacto	nvarchar(100)	<input type="checkbox"/>

Ilustración 44: Tabla Otros_datospersonales

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece el "otro dato personal".
- Un identificador que será el tipo de contacto de la tabla Contacto.
- Un campo para incluir el valor del tipo de contacto.

La función de esta tabla es la de guardar los distintos valores de cada contacto. Un ejemplo sería:

- Tipo de contacto: Infojobs
- Contacto: nombre apellido1 apellido2 en Infojobs.

Para un mismo tipo de contacto, podemos tener una o varias personas. En esta tabla guardaremos la persona del CV, con el `id_tipo_contaco` sacaremos el tipo de contacto que tiene y con el campo `contacto` incluiremos su valor concreto. Los tipos de contacto pueden ser Twitter, Facebook, Infojobs, algo que es fijo y no cambia, por lo que están guardados en la tabla Contacto y nunca se modifican. En esta tabla se obtiene ese tipo de contacto para luego exponer el valor específico de cada persona. Los identificadores son de tipo guid y todos los campos son obligatorios. El campo `contacto` es un string de 100 caracteres como máximo.

Estudios

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
periodo	nvarchar(100)	<input type="checkbox"/>
titulo	nvarchar(200)	<input type="checkbox"/>
especialidad	nvarchar(200)	<input checked="" type="checkbox"/>
centro	nvarchar(200)	<input checked="" type="checkbox"/>

Ilustración 45: Tabla Estudios

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece cada estudio.
- Un campo para incluir el periodo de realización de ese estudio.
- Un campo para incluir el nombre del título obtenido en ese estudio.
- Un campo para incluir la especialidad sobre ese estudio.
- Un campo para incluir el centro donde se ha realizado el estudio.

La función de esta tabla es la de almacenar todos los estudios de cada persona. La especialidad y el centro no serán campos obligatorios. Los identificadores son de tipo guid, el periodo es un string de 100 caracteres de máximo y el titulo, la especialidad y el centro de 200 caracteres.

Otros_titulos

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
fecha_obtencion	nvarchar(50)	<input type="checkbox"/>
nombre_titulo	nvarchar(200)	<input type="checkbox"/>
centro	nvarchar(100)	<input checked="" type="checkbox"/>

Ilustración 46: Tabla Otros_titulos

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece el título en concreto.
- Un campo para incluir la fecha de obtención del título.
- Un campo para incluir el nombre del título.
- Un campo para incluir el centro donde se obtuvo el título.

La función de esta tabla es la de guardar títulos obtenidos que estén fuera de estudios reglados. Los identificadores son de tipo guid y todos los campos son obligatorios excepto el centro. Los demás campos son de tipo string, donde fecha de obtención es de 50 caracteres, nombre título de 200 y el centro de 100.

Otros_cursos

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
periodo	nvarchar(50)	<input type="checkbox"/>
nombre_curso	nvarchar(200)	<input type="checkbox"/>
centro	nvarchar(100)	<input type="checkbox"/>

Ilustración 47: Tabla Otros_cursos

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece el curso en concreto.
- Un campo para incluir el periodo mientras la persona realizaba o realiza el curso.
- Un campo para incluir el nombre del curso.
- Un campo para incluir el centro donde se realizó el curso.

La función de esta tabla es la de guardar cursos realizados que estén fuera de estudios reglados. Los identificadores son de tipo guid y todos los campos son obligatorios. Los demás campos son de tipo string, donde periodo es de 50 caracteres, nombre curso de 200 y el centro de 100.

Experiencia

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
periodo	nvarchar(50)	<input type="checkbox"/>
puesto	nvarchar(200)	<input type="checkbox"/>
empresa	nvarchar(200)	<input type="checkbox"/>
recomendaciones	nvarchar(500)	<input checked="" type="checkbox"/>
nivel	nvarchar(50)	<input checked="" type="checkbox"/>
sector_empresa	nvarchar(50)	<input checked="" type="checkbox"/>
salario	nvarchar(50)	<input checked="" type="checkbox"/>
personal_cargo	nvarchar(50)	<input checked="" type="checkbox"/>
descripcion_puesto	nvarchar(3000)	<input checked="" type="checkbox"/>

Ilustración 48: Tabla Experiencia

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece la experiencia en concreto.
- Un campo para incluir el periodo de duración de la experiencia.
- Un campo para incluir el puesto ocupado en la experiencia.
- Un campo para incluir la empresa donde se ha realizado la experiencia.
- Un campo para incluir las recomendaciones en esa experiencia.
- Un campo para incluir el nivel adquirido en esa experiencia.
- Un campo para incluir el sector de la empresa ocupado en esa experiencia.
- Un campo para incluir el salario obtenido durante la experiencia.
- Un campo para incluir el personal a cargo tenido durante la experiencia.
- Un campo para incluir la descripción del puesto realizado durante la experiencia.

La función de esta tabla consiste en guardar las experiencias profesionales de cada persona. Los identificadores son de tipo guid. Sólo el periodo, el puesto y la empresa serán campos obligatorios. Excepto los identificadores, todos los campos son strings, donde el periodo, nivel, sector de empresa, salario y personal a cargo tienen 50 caracteres de máximo, puesto y empresa 200, recomendaciones 500 y descripción del puesto 3000.

Conocimientos_tecnologias

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
conocimiento	nvarchar(500)	<input type="checkbox"/>
id_experiencia	uniqueidentifier	<input checked="" type="checkbox"/>
nivel_conocimiento	nvarchar(100)	<input checked="" type="checkbox"/>

Ilustración 49: Tabla Conocimientos_tecnologias

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece el conocimiento en concreto.
- Un campo para incluir el conocimiento de la experiencia profesional.
- Un identificador de la experiencia donde se aplican los conocimientos.
- Un campo para incluir el nivel del conocimiento que tiene la persona.

La función de esta tabla es la de guardar los conocimientos que tiene cada persona en cada experiencia profesional. Es exactamente lo mismo que ocurría con las categorías, pues en una misma experiencia profesional, una persona puede tener un conocimiento o varios. Esta tabla resuelve ese problema de 1 a n. Además, esta tabla permite guardar los conocimientos de las personas fuera de experiencias profesionales, ya sea por cursos, estudios, o cualquier otro método. Hay un apartado de "Otros conocimientos" en el que las personas rellenan con conocimientos que no han adquirido en una experiencia profesional. En caso de insertar conocimientos sin experiencia, el id de la experiencia será null, por eso permite valores vacíos. Los identificadores son tipo guid, el conocimiento un string de 500 caracteres y el nivel otro string de 100 caracteres.

Idiomas

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
idioma	nvarchar(50)	<input type="checkbox"/>
hablado	nvarchar(50)	<input type="checkbox"/>
escrito	nvarchar(50)	<input type="checkbox"/>
leido	nvarchar(50)	<input checked="" type="checkbox"/>
comentarios	nvarchar(500)	<input checked="" type="checkbox"/>

Ilustración 50: Tabla Idiomas

Esta tabla se compone de lo siguiente:

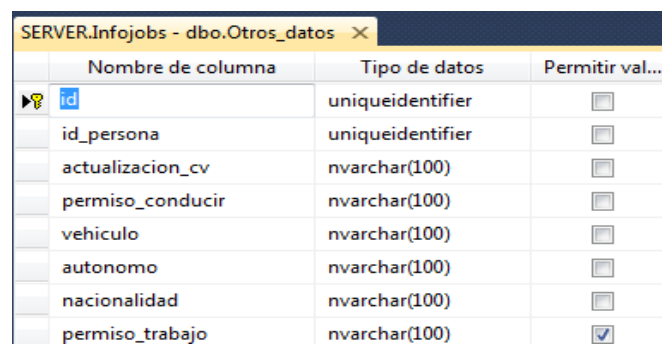
- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece el idioma en concreto.

- Un campo para incluir el idioma.
- Un campo para incluir el nivel hablado del idioma.
- Un campo para incluir el nivel escrito del idioma.
- Un campo para incluir el nivel leído del idioma.
- Un campo para incluir comentarios sobre el idioma.

La función de esta tabla es la de guardar el idioma de cada persona. Los identificadores son de tipo guid, y sólo el campo leído y comentarios podrán ser nulos. Excepto los identificadores, todos los campos son strings de 50 caracteres máximo excepto comentarios que tiene 500.

Otros_datos

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
actualizacion_cv	nvarchar(100)	<input type="checkbox"/>
permiso_conducir	nvarchar(100)	<input type="checkbox"/>
vehiculo	nvarchar(100)	<input type="checkbox"/>
autonomo	nvarchar(100)	<input type="checkbox"/>
nacionalidad	nvarchar(100)	<input type="checkbox"/>
permiso_trabajo	nvarchar(100)	<input checked="" type="checkbox"/>

Ilustración 51: Tabla Otros_datos

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenecen los otros datos.
- Un campo para incluir la fecha de la última actualización del CV.
- Un campo para incluir si la persona tiene o no permiso de conducir.
- Un campo para incluir si la persona tiene o no vehículo propio.
- Un campo para incluir si la persona es autónoma o no.
- Un campo para incluir la nacionalidad de la persona.
- Un campo para incluir el permiso de trabajo de la persona.

La función de esta tabla es la de guardar información adicional sobre la persona, más allá de sus datos personales, como una complementación. Los identificadores son de tipo guid y sólo el permiso de trabajo podrá ser nulo. Excepto los identificadores, todos los campos son de tipo string con 100 caracteres de máximo.

Futuro_empleo

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_persona	uniqueidentifier	<input type="checkbox"/>
puesto_preferido	nvarchar(500)	<input checked="" type="checkbox"/>
categoria_preferida	nvarchar(500)	<input checked="" type="checkbox"/>
situacion_actual	nvarchar(500)	<input checked="" type="checkbox"/>
motivo	nvarchar(500)	<input checked="" type="checkbox"/>
objetivo	nvarchar(1000)	<input checked="" type="checkbox"/>
salario_minimo_aceptado	nvarchar(500)	<input checked="" type="checkbox"/>
salario_deseado	nvarchar(500)	<input checked="" type="checkbox"/>
contrato_preferido	nvarchar(500)	<input checked="" type="checkbox"/>
jornada_laboral_preferida	nvarchar(500)	<input checked="" type="checkbox"/>
cambio_residencia	nvarchar(500)	<input checked="" type="checkbox"/>
disponibilidad_viajar	nvarchar(500)	<input checked="" type="checkbox"/>
destino_preferido	nvarchar(500)	<input checked="" type="checkbox"/>

Ilustración 52: Tabla Futuro_empleo

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la persona a la que pertenece los datos de un posible futuro empleo.
- Un campo para incluir el puesto preferido en el futuro empleo.
- Un campo para incluir la categoría preferida en el futuro empleo.
- Un campo para incluir la situación actual.
- Un campo para incluir el motivo de la búsqueda de un futuro empleo.
- Un campo para incluir el objetivo de la búsqueda de un futuro empleo.
- Un campo para incluir el salario mínimo que se aceptaría en el futuro empleo.
- Un campo para incluir el salario deseado que se querría en el futuro empleo.
- Un campo para incluir el contrato preferido que se querría en el futuro empleo.
- Un campo para incluir la jornada laboral que se querría en el futuro empleo.
- Un campo para incluir si hay posibilidad de cambio de residencia en caso de encontrar un futuro empleo.
- Un campo para incluir si hay disponibilidad para viajar en caso de encontrar un futuro empleo.
- Un campo para incluir el destino preferido en caso de tener disponibilidad para viajar.

La función de esta tabla consiste en guardar una serie de datos requeridos a la hora de obtener información sobre un posible futuro empleo para la persona del CV. Los identificadores son de tipo guid, y todos los demás campos pueden ser nulos. Todos los campos son strings de 500 caracteres de capacidad máxima excepto objetivo, que tiene 1000.

Peticion

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir null
id	uniqueidentifier	<input checked="" type="checkbox"/>
nombre	nvarchar(100)	<input checked="" type="checkbox"/>
nombre_solicitante	nvarchar(100)	<input checked="" type="checkbox"/>
detalle_solicitante	nvarchar(100)	<input checked="" type="checkbox"/>
fecha_solicitante	nvarchar(100)	<input checked="" type="checkbox"/>
empresa	nvarchar(100)	<input checked="" type="checkbox"/>
proyecto	nvarchar(100)	<input checked="" type="checkbox"/>
lugar	nvarchar(100)	<input checked="" type="checkbox"/>
duracion	nvarchar(100)	<input checked="" type="checkbox"/>
fecha_ini	nvarchar(100)	<input checked="" type="checkbox"/>
fecha_fin	nvarchar(100)	<input checked="" type="checkbox"/>
tipo_contratacion	nvarchar(100)	<input checked="" type="checkbox"/>
descuentos	nvarchar(100)	<input checked="" type="checkbox"/>
observaciones	nvarchar(100)	<input checked="" type="checkbox"/>
horario	nvarchar(100)	<input checked="" type="checkbox"/>
titulacion_requerida	nvarchar(100)	<input checked="" type="checkbox"/>
titulacion_deseable	nvarchar(100)	<input checked="" type="checkbox"/>
perfil_requerido	nvarchar(100)	<input checked="" type="checkbox"/>
exp_requerido	nvarchar(100)	<input checked="" type="checkbox"/>
perfil_deseable	nvarchar(100)	<input checked="" type="checkbox"/>
exp_deseable	nvarchar(100)	<input checked="" type="checkbox"/>
sueldo	nvarchar(100)	<input checked="" type="checkbox"/>
estado	nvarchar(100)	<input checked="" type="checkbox"/>

Ilustración 53: Tabla Peticion

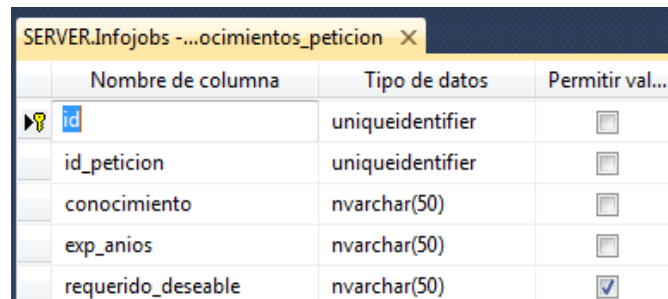
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir el nombre de la petición.
- Un campo para incluir el nombre del solicitante de la petición.
- Un campo para incluir detalles sobre la petición.
- Un campo para incluir la fecha de creación de la petición.
- Un campo para incluir la empresa de la petición.
- Un campo para incluir el proyecto de la petición.
- Un campo para incluir el lugar de la petición.
- Un campo para incluir la duración de la petición.
- Un campo para incluir la fecha de inicio de la petición.
- Un campo para incluir la fecha de fin de la petición.
- Un campo para incluir el tipo de contratación.
- Un campo para incluir los descuentos posibles en la contratación.
- Un campo para incluir observaciones.
- Un campo para incluir un horario.
- Un campo para incluir la titulación requerida en la petición.
- Un campo para incluir la titulación deseable en la petición.
- Un campo para incluir el perfil requerido de la petición.
- Un campo para incluir la experiencia requerida con ese perfil requerido.
- Un campo para incluir el perfil deseable de la petición.
- Un campo para incluir la experiencia deseable con ese perfil deseable.
- Un campo para incluir el sueldo de la petición.
- Un campo para incluir el estado de la petición.

La función de esta tabla es la de guardar todas las peticiones que se realicen en el sistema. El identificador es de tipo guid y sólo el nombre de la petición, del solicitante, la fecha, la empresa y el lugar son obligatorios. Excepto el identificador, todos los campos son de tipo string con 100 caracteres de máximo.

Conocimientos_peticion

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_peticion	uniqueidentifier	<input type="checkbox"/>
conocimiento	nvarchar(50)	<input type="checkbox"/>
exp_anios	nvarchar(50)	<input type="checkbox"/>
requerido_deseable	nvarchar(50)	<input checked="" type="checkbox"/>

Ilustración 54: Tabla Conocimientos_peticion

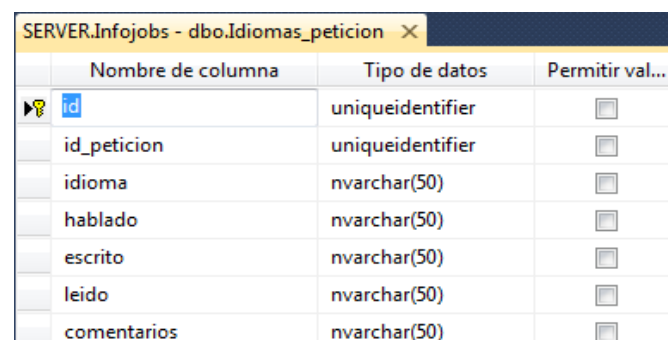
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la petición a la que pertenece el conocimiento en concreto.
- Un campo para incluir el conocimiento de la petición.
- Un campo para incluir los años de experiencia que se tiene sobre ese conocimiento.
- Un campo para incluir si el conocimiento es requerido expresamente, o si es deseable pero no obligatorio.

Esta tabla tiene como función guardar los conocimientos que se piden en una petición para que sean comprobados en los CVS asociados a las peticiones. Estos conocimientos deben estar incluidos en los CVS para que puedan ser estudiados. Los que no cumplan los requisitos de la petición, se descartan o no se asocian a la petición. En este caso todos los campos son obligatorios excepto si es un conocimiento requerido o deseable.

Idiomas_peticion

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_peticion	uniqueidentifier	<input type="checkbox"/>
idioma	nvarchar(50)	<input type="checkbox"/>
hablado	nvarchar(50)	<input type="checkbox"/>
escrito	nvarchar(50)	<input type="checkbox"/>
leido	nvarchar(50)	<input type="checkbox"/>
comentarios	nvarchar(50)	<input type="checkbox"/>

Ilustración 55: Tabla Idiomas_peticion

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador de la petición a la que pertenece el idioma en concreto.
- Un campo para incluir el idioma.
- Un campo para incluir el nivel hablado del idioma.

- Un campo para incluir el nivel escrito del idioma.
- Un campo para incluir el nivel leído del idioma.
- Un campo para incluir comentarios sobre el idioma.

La función de esta tabla es la de guardar el idioma de una petición. Los identificadores son de tipo guid, y todos los campos son obligatorios. Excepto los identificadores, todos los campos son strings de 50 caracteres máximo.

CV_peticion

La forma de la tabla se puede ver en la siguiente ilustración:

Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
id_cv	uniqueidentifier	<input type="checkbox"/>
id_peticion	uniqueidentifier	<input type="checkbox"/>
fecha_asociacion	nvarchar(50)	<input checked="" type="checkbox"/>
estado	nvarchar(50)	<input checked="" type="checkbox"/>

Ilustración 56: Tabla CV_peticion

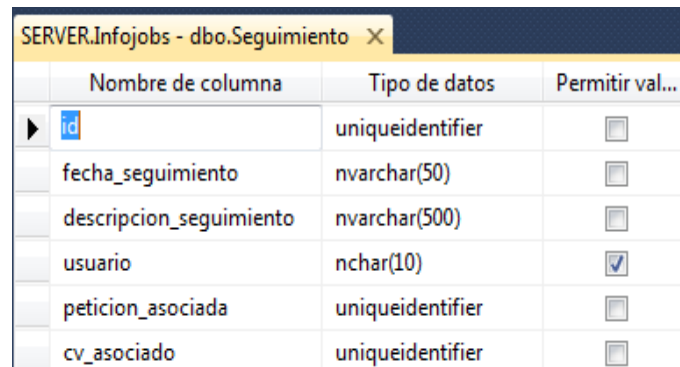
Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un identificador del currículm asociado a una petición.
- Un identificador de la petición, que es la que contiene los CVS asociados.
- Un campo incluir la fecha en la que se realiza la asociación del CV a la petición.
- Un campo estado, que indicará si la petición es nueva, es decir, se acaba de crear, si es abierta, esto es que se está modificando, o si es cerrada, en cuyo caso la petición habrá sido guardada.

La función de esta tabla reside en que dentro de una petición podemos tener un CV o varios asociados. Se produce un problema de 1 a n, que se resuelve con esta tabla intermedia. Con esta tabla sabremos que CVS están asociados con qué peticiones, la fecha en la que se hizo y como se encuentra la petición. Lo importante es saber que CVS están en cada petición, por lo que sólo serán no obligatorios los campos de fecha y de estado, ambos de tipo string con capacidad de 50 caracteres. Los dos id son de tipo guid.

Seguimiento

La forma de la tabla se puede ver en la siguiente ilustración:



Nombre de columna	Tipo de datos	Permitir val...
id	uniqueidentifier	<input type="checkbox"/>
fecha_seguimiento	nvarchar(50)	<input type="checkbox"/>
descripcion_seguimiento	nvarchar(500)	<input type="checkbox"/>
usuario	nchar(10)	<input checked="" type="checkbox"/>
peticion_asociada	uniqueidentifier	<input type="checkbox"/>
cv_asociado	uniqueidentifier	<input type="checkbox"/>

Ilustración 57: Tabla Seguimiento

Esta tabla se compone de lo siguiente:

- Un identificador único para cada tupla de la tabla.
- Un campo para incluir la fecha en la que se realiza un seguimiento.
- Un campo para incluir la descripción de un seguimiento.
- Un campo para incluir el usuario que realiza el seguimiento.
- Un identificador de la petición a la que pertenece el seguimiento.
- Un identificador del CV sobre el que se realiza el seguimiento.

La función de esta tabla es la de guardar los seguimientos que se realizan sobre CVS asociados a peticiones. Los identificadores son de tipo guid y solo el usuario puede ser nulo. Fecha de seguimiento es un string de 50 caracteres, descripción de 500 y usuario de 10.

Análisis

Esta es la base de datos que da soporte al sistema y que se encarga de guardar la información de los CVS, ya sean importados directamente o creados a mano. A continuación se va a hacer un pequeño análisis general de lo explicado anteriormente.

En primer lugar disponemos de varias tablas que son similares en cuanto a la finalidad. Estas tablas son:

- Conocimientos_mantenimiento
- Contacto
- Idiomas_general
- Paises
- Perfiles
- Provincias
- Titulaciones

Todas estas tablas comparten una misma finalidad. Están creadas para realizar una inserción inicial en ellas, y en principio no modificarlas más. Esto es porque los datos introducidos en ellas no van a cambiar, existen así y tienen que ser así. Un conocimiento como por ejemplo "Java", un contacto como por ejemplo "Twitter", el idioma "Inglés", el país "Suecia"... todos estos valores no van a cambiar. Todas estas tablas servirán para dar soporte a otras tablas, y ese soporte se verá reflejado en la página web.

El soporte es el siguiente. En caso de estar dando de alta a un candidato, podemos querer crear un estudio. A la hora de crearlo, dispondremos de un select (una entrada donde pinchar y se abre un desplegable con una serie de opciones) y este select tendrá como opciones los valores de la tabla titulaciones. La siguiente captura es un ejemplo:

The screenshot shows a web form titled 'ESTUDIOS'. It includes fields for 'Fecha inicio', 'Fecha fin', 'Especialidad', and 'Centro'. A dropdown menu labeled 'Seleccionar titulación' is open, displaying a list of educational titles: 'Selección de titulación', 'Bachillerato', 'Ciclo Formativo Grado Superior', 'Diplomado', 'Educación Secundaria Obligatoria', 'Formación Profesional Grado Medio', 'Formación Profesional Grado Superior', 'Grado', 'Ingeniero Superior', 'Ingeniero Técnico', 'Máster', and 'Postgrado'. Buttons for 'Quitar Estudio 1', 'Añadir Estudio 2', and 'Añadir Título 1' are also visible.

Ilustración 58: Ejemplo de select

Lo mismo ocurre por ejemplo al crear un usuario, ya que tendremos que darle un perfil determinado (administrador, técnico...) y estos valores vendrán dados por la tabla Perfiles. Esto ocurrirá al dar de alta un candidato, a la hora de editarlo, al crear una petición...etc.

No obstante, algunas de estas tablas no serán siempre fijas. Por ejemplo, en el caso de Provincias, siempre será fija ya que están todas las provincias de España. Sin embargo, para Conocimientos_mantenimiento, es posible que alguien pueda tener un conocimiento que no estuviera en la lista, ya que hay demasiados conocimientos. Por tanto, se dará la opción de escribir un conocimiento manual, además del select previamente dicho. En caso de escribirlo manual, si existe en la base de datos, se rellenará automáticamente el select con ese conocimiento. En caso de que no exista, el conocimiento nuevo se insertará en la base de datos, y aparecerá en el select para el futuro. La siguiente captura muestra un ejemplo de ello:

The screenshot shows a web form titled 'OTROS CONOCIMIENTOS'. It features two input fields for knowledge. 'Conocimiento 1' has a dropdown menu with 'Apache' selected. 'Conocimiento 2' has a dropdown menu with 'Introducir otro...' selected and a text input field containing 'Java'. Buttons for 'Quitar Conocimiento 1', 'Quitar Conocimiento 2', and 'Añadir Conocimiento 3' are present.

Ilustración 59: Ejemplo de inserción de conocimientos

Después tenemos la tabla Usuarios, la cual simplemente se encargará de almacenar los distintos usuarios del sistema, que tendrán cada uno sus datos propios para acceder al sistema y unos privilegios definidos en el perfil.

Llegamos a la tabla Datos_personales, que es la más importante de todas. En esta tabla se hace la inserción normal de los datos personales de una persona como su nombre o su email, pero la esencia está en el identificador que se crea para cada tupla de esta tabla. A la hora de insertar estudios, títulos, experiencias...todo gira en torno a la persona que realiza esas acciones o tiene esos conocimientos. Por tanto, el identificador de la tabla Datos_personales será necesario en todas las tablas que requieran identificar a una persona.

Estas tablas son:

- Otros_datospersonales
- Estudios
- Otros_titulos
- Otros_cursos
- Experiencia
- Idiomas
- Otros_datos
- Futuro_empleo
- CV_peticion

Todas estas tablas heredan el identificador de la persona incluido en la tabla Datos_personales. A su vez, cada tabla independientemente puede obtener datos de las tablas explicadas justo antes, como por ejemplo Estudios, que cogerá valores de la tabla Titulaciones.

Por otro lado tenemos la tabla Conocimientos_tecnologias, que depende de la tabla Experiencia. Una experiencia puede contener uno o varios conocimientos. Todo no se puede almacenar en una sola tupla, por tanto, en la tabla Conocimientos_ tecnologías hay un campo de identificador de experiencia y de persona, para asociar a una persona, en una experiencia, tantos conocimientos como tenga. Además, la tabla conocimientos tecnologías sirve para guardar conocimientos fuera de experiencias, como otros conocimientos a parte. Al insertar conocimientos en esta tabla, se hará una comprobación con Conocimientos_mantenimiento y en caso de no existir el conocimiento, se insertaría en ambas.

Por último tenemos las peticiones, que es el segundo elemento fundamental después de Datos_personales. Al crear la petición, se pueden realizar muchas acciones sobre ella, y las siguientes tablas dependen de ello:

- Conocimientos_peticion
- Idiomas_peticion
- CV_peticion
- Seguimiento

Todas estas tablas heredan el identificador de la petición incluido en la tabla Peticion. A su vez, cada tabla independientemente puede obtener datos de las tablas explicadas justo antes, como por ejemplo CV_peticion, que cogerá el identificador del CV de la tabla Datos_personales. Una vez desarrollada la base de datos, vamos a explicar el desarrollo del programa fundamental.

5.2. Programa de recogida de información

El programa, como ya he indicado anteriormente, se encarga de leer un CV en formato html que haya sido descargado a través del portal web Infojobs o de Tecnoempleo.

En primer lugar, viene la lectura del fichero. En la página web, hay una página que se llama "Inicio-importar.php", que contiene un apartado de selección de archivos y una opción de enviar, que aparece con un botón "Enviar fichero".

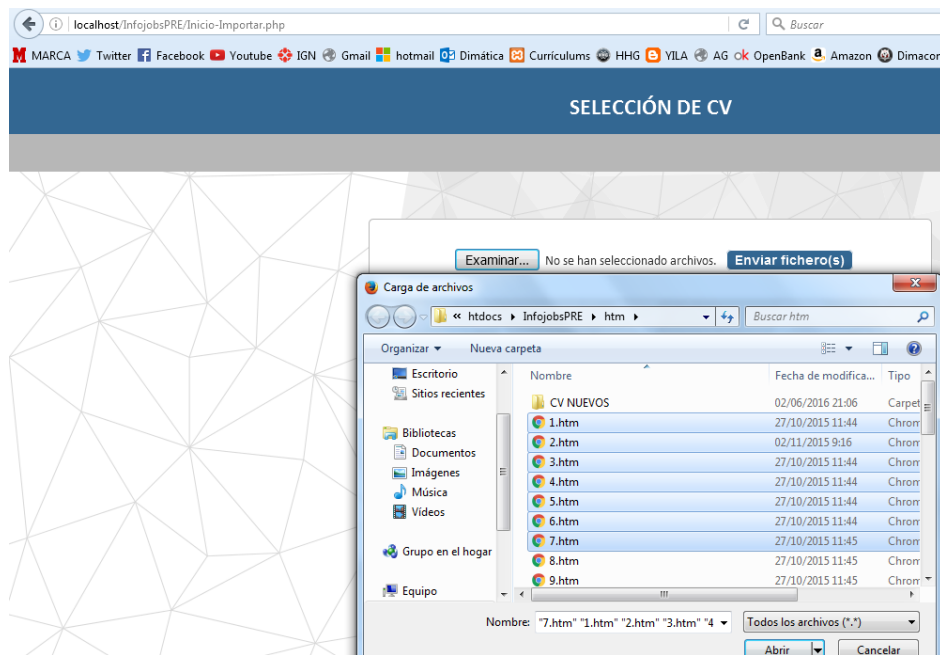


Ilustración 60: Ejemplo de importar archivos

Al seleccionar tantos archivos como queramos y pinchar en el botón de enviar, se ejecutará un archivo llamado "nuevo.php" que es el programa en cuestión.

Al ejecutarse "nuevo.php", lo primero que se hace es un bucle for para recorrer todos los archivos que se han enviado (como se explica en el apartado 4. Análisis - 4.3. Programa de recogida de información). Se guarda la ruta del archivo y el nombre en variables, y se pasa a llamar a una función llamada comprobarFormato.

```
function comprobarFormato($file){
    $html = new simple_html_dom();
    $html = file_get_html($file);

    if(!is_null($variableComprobar = $html->find('title',0)->plaintext)){
        $comprobacion = strpos($variableComprobar, 'Tecnoempleo');
        if($comprobacion !== false){
            insertarTecnoempleo($file);
        }
        else{
            insertarInfojobs($file);
        }
    }
}
```

Ilustración 61: Función comprobarFormato()

Esta función se encargará de distinguir el tipo de CV que ha sido seleccionado, es decir, reconocer si el CV introducido es de Infojobs o de Tecnoempleo. En primer lugar se crea un objeto llamado "html" (de la librería Simple html dom, que permite con PHP acceder a etiquetas html de los documentos) que guarda el CV introducido. Ahora, realizamos una

comprobación, y es que buscamos la palabra "Tecnoempleo" dentro del objeto "html" en la primera etiqueta title que aparezca (find('title',0)). En caso de que se encuentre la palabra, la variable "comprobacion" valdrá true, y false en caso contrario. Dependiendo de si es de un tipo u otro, se llamará a la función insertarInfojobs o a la función insertarTecnoempleo.

Ahora, vamos a analizar la función insertarInfojobs por dentro. Esta función se ejecutará en caso de que la variable "comprobacion" valga false. En primer lugar, tenemos que crear otro objeto "html", ya que es necesario guardar el CV que estamos tratando. Una vez creado, lo primero que hay que hacer es acceder a las etiquetas que no se repitan y sean únicas, para obtener su valor. La siguiente captura es un ejemplo de ello:

```
//DATOS PERSONALES
$nombre_completo = $html->find('div[id=idNombreCompleto]',0)->plaintext;
if($nombre_completo==null){
    $nombre_completo = $html->find('span[id=idNombreCompleto]',0)->plaintext;
}
$email = $html->find('div[id=idEmail]',0)->plaintext;
$sexo = $html->find('div[id=idSexo]',0)->plaintext;
$fecha_nacimiento = $html->find('div[id=idFechaDeNacimiento]',0)->plaintext;
$direccion = $html->find('div[id=idDireccion]',0)->plaintext;
$codigo_postal = $html->find('div[id=idCodigoPostal]',0)->plaintext;
$poblacion = $html->find('div[id=idPrefijoPoblacion]',0)->plaintext;
$provincia = $html->find('div[id=idProvincia]',0)->plaintext;
$snif = $html->find('div[id=idDNI]',0)->plaintext;
$telefono_fijo = $html->find('div[id=idTelefonoFijo]',0)->plaintext;
$telefono_movil = $html->find('div[id=idTelefonoMovil]',0)->plaintext;
$telefono_extranjero = $html->find('div[id=idTelefonoExtranjero]',0)->plaintext;
```

Ilustración 62: Ejemplo recogida datos personales de infojobs

Sabemos que el nombre, el e-mail, el sexo, la fecha de nacimiento...etc. no van a repetirse en el CV, son únicos. Por tanto, accedemos a cada una con un identificador específico. Sabemos por ejemplo que existe un div cuyo id es "idEmail", y este id no se repite más en todo el CV, por lo que sabemos que en caso de encontrar un div cuyo identificador sea "idEmail" contiene el e-mail de la persona. Por tanto, guardamos en la variable "email" el texto (plaintext) que haya dentro del primer div (0) cuyo identificador sea "idEmail". Una vez hecho esto, ya tenemos guardado en la variable email, el email de la persona en cuestión. Este proceso es el mismo para todos los valores del CV que sea únicos y no se repitan.

Sin embargo, hay valores que se repiten, como los estudios, las experiencias o los idiomas entre otros, y hay que tratarlos de una manera especial.

Lo veremos todo más claro con la siguiente captura:

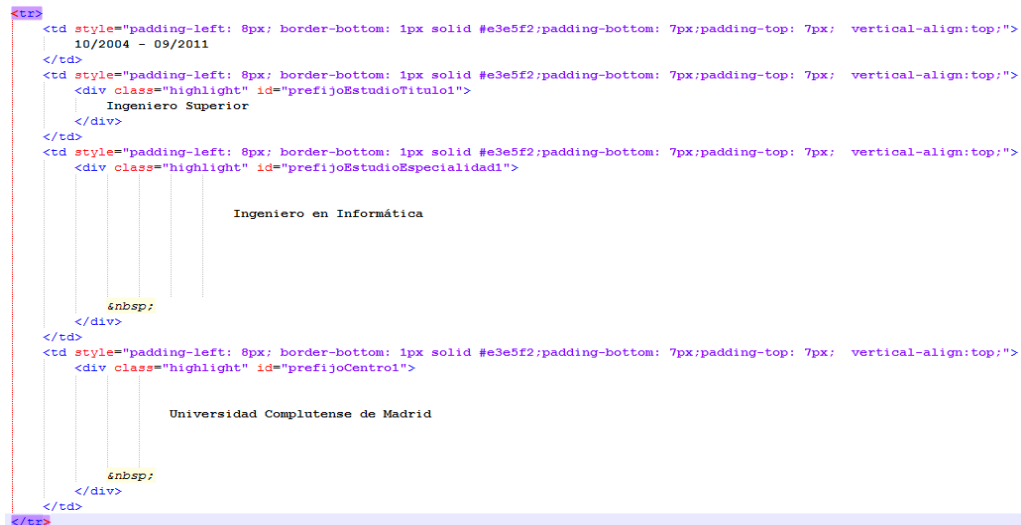


Ilustración 63: Ejemplo de estudios de infojobs

En la captura podemos ver un ejemplo de un estudio de una persona cualquiera, que está incluido dentro de una etiqueta `<tr>` `</tr>` que está marcada. Vemos que lo primero que aparece en negrita es un periodo del estudio, después el título del estudio, luego la especialidad del estudio y finalmente el centro donde realizó el estudio. Vemos como esta información no está contenida dentro de etiquetas con identificadores únicos, y además, que puede ser que la persona tenga varios estudios, por lo que la estructura de la captura se repetiría por todos los estudios que tuviera. Hay que hacer un programa recursivo que detecte todos los estudios que pudiera haber, y pare cuando ya no haya más. La siguiente captura lo muestra:

```
//ESTUDIOS
$controlador=1;
$estudios=$html->find('div[id=linkStudies]',0);
if($estudios!=null){
    $clase=$estudios->find('table',0);
    foreach($clase->find('tr') as $tr){
        $elemento = $tr->style;
        if($elemento==''){
            foreach($tr->find('td') as $td){
                ${"resultado".$controlador} = $td->plaintext;
                $controlador++;
            }
            $periodo=controlar($resultado1);
            $titulo=controlar($resultado2);
            $especialidad=controlar($resultado3);
            $centro=controlar($resultado4);
            //Insercion en tabla Estudios
            $id3 = NewGuid();
            $sql_estudios = "INSERT INTO Estudios(id,id_persona, periodo, titulo, especialidad, centro)
            VALUES ('".$id3."','".$id."','".$periodo."','".$titulo."','".$especialidad."','".$centro."')";
            $stmtEstudios = $db->exec($conn, $sql_estudios);
            $db->free_result($stmtEstudios);
            $controlador=1;
        }
    }
}
```

Ilustración 64: Ejemplo de recursividad en estudios de infojobs

En primer lugar, buscamos el identificador "linkStudies", que nos dice que esa persona tiene estudios. Si existe, entonces la variable estudios tendrá el valor del div cuyo identificador sea "linkStudies". Y aquí viene algo importante, que es la siguiente línea:

```
$clase=$estudios->find('table',0);
```

Al igual que para acceder a "linkStudies" utilizábamos el objeto "html" creado al inicio, al crear la variable estudios que tiene como resultado el div que contiene ese identificador, podemos acceder ahora a lo que haya dentro únicamente de ese div. Digamos que la variable estudios ahora se corresponde con el div cuyo identificador es "linkStudies", por lo que si en lugar de utilizar la variable "html" que contenía el CV entero utilizamos "estudios", estaremos accediendo solamente a elementos del interior de ese div. Por tanto, guardamos en la variable "clase" la primera tabla que aparezca (find('table',0)) dentro de la variable estudios. Una vez hecho esto, tenemos en la variable clase el contenido de la primera tabla del div cuyo identificador es "linkStudies". Ahora, una vez accedido a esa tabla, tendremos que recorrer todas las etiquetas <tr> que haya, ya que estas se corresponden con las etiquetas marcadas en la captura anterior. Eso lo hacemos con un foreach, que se encargará de recorrer todos los tr que haya dentro de clase, y a su vez, dentro de cada tr, guardará el texto que haya dentro de cada td. Esto lo guardará en cuatro variables diferentes, una para cada campo, y realizará la inserción de la información en la base de datos. Una vez hecha la inserción, pasará a buscar otro tr, y en caso de que exista, repetirá el proceso. Así hasta que ya no haya tr dentro de la primera tabla del div con identificador "linkStudies", es decir, que no haya más estudios.

Este proceso es muy útil, ya que con pocas líneas de código, se almacena grandes cantidades de información, y es válido para todos los CVS que sean de Infojobs. Es un proceso que puede parecer algo enrevesado, pero la mecánica es muy clara y simple. Hay que acceder a elementos que se repitan y sean iguales en todos los CVS, ya sean tablas, divs, tr, o lo que sea, y mediante un foreach recorrer todos los que haya haciendo las inserciones oportunas. Este proceso es igual para experiencia, para idiomas, para conocimientos, y para todo lo que sea recursivo y se pueda repetir. Lógicamente, la manera de obtener los datos es diferente, ya que para estudios comprobamos la existencia de un div con identificador "linkStudies", mientras que para experiencia sería "linkExperience", pero la mecánica es la misma exactamente. Todo este proceso hace referencia a cómo obtener los datos del CV, que hemos visto que hay dos maneras, o directamente mediante identificadores únicos, o recursivamente buscando patrones que se repiten. Ahora vamos a explicar cómo se realizan las inserciones a la base de datos.

En primer lugar, hay que realizar una conexión con la base de datos. Tanto en este programa como en numerosas páginas, si es necesario obtener información de la base de datos, modificarla o insertar una nueva, se hace lo siguiente:

```
$conn = conectarBD();
```

Ilustración 65: Variable conn

Esto crea una variable "con" que contiene el resultado de la función conectarBD, que está incluida en el archivo funciones.php, y que todos los archivos tienen incluido al inicio mediante (entre otros includes):

```
<?php
session_start();
if($_SESSION["usuario"]==""){
    header("Location:/InfojobsPRE/Index.php");
}
header("Content-Type: text/html;charset=utf-8");
include_once('globals.php');
include('funciones.php');
include_once('simple_html_dom.php');
error_reporting (E_ALL ^ E_NOTICE);
```

Ilustración 66: Ejemplo de includes

Por tanto, dentro del archivo funciones.php tenemos la función creada, que es la siguiente:

```
function conectarBD(){
    $conn = odbc_connect(nombreBBDD, usuarioBBDD, passBBDD);
    if($conn) {
        //echo "Conexión establecida.<br />";
        //echo "<br>";
    }else{
        echo "Conexión no se pudo establecer.<br />";
        echo "<br>";
        die(print_r( sqlsrv_errors(), true));
    }
    return $conn;
}
```

Ilustración 67: Función conectarBD()

Esta función devuelve un identificador de conexión en caso de éxito, y un mensaje de error en caso de fracaso. Para poder hacer la conexión, se utilizan tres variables globales, que están incluidas dentro del archivo globals.php, y que incluyen en este caso todos los demás archivos. Las variables son las siguientes:

```
define("nombreBBDD","ODBC_Infojobs");
define("usuarioBBDD","");
define("passBBDD","");
```

Ilustración 68: Valores globales

Estos valores siempre son los mismos, y definen el conector entre el Microsoft SQL y el servidor, el usuario y la contraseña para conectar.

Una vez realizada la conexión, y teniendo los valores correspondientes, simplemente tenemos que crear la SQL que realice una inserción, y ejecutarla con el comando de PHP odbc_exec.

Por último habría que liberar la conexión, para que no se colapse con otras conexiones, y para ello utilizo el comando `odbc_free_result` pasándole como parámetro el resultado de la ejecución anterior.

```
//Insercion en tabla Estudios
$id3 = NewGuid();
$sql_estudios = "INSERT INTO Estudios(id,id_persona, periodo, titulo, especialidad, centro)
VALUES ('".$id3."','".$id."','".$periodo."','".$titulo."','".$especialidad."','".$centro."')";
$stmtEstudios = odbc_exec($conn, $sql_estudios);
odbc_free_result($stmtEstudios);
```

Ilustración 69: Ejemplo de inserción en base de datos

Todo esto ha sido explicado para la función `insertarInfojobs`, pero para el caso de `insertarTecnoempleo` es exactamente el mismo. Los datos se obtienen de las dos mismas maneras, o por identificadores únicos o por recursividad, y las inserciones a la base de datos se hacen de la misma manera.

Por último, existe la posibilidad de que se quiera insertar un mismo CV dos veces, o que se quiera insertar un CV de una persona que ya existe en la base de datos.

```
$comprobarDNI = "SELECT * FROM Datos_personales WHERE NIF = '".$nif."'";
$stmtcomprobar = odbc_exec($conn, $comprobarDNI);
if(odbc_fetch_row($stmtcomprobar)){
    $result = odbc_result($stmtcomprobar, 'NIF');
}
if($result!=null){
    $sql = "SELECT id FROM Datos_personales WHERE NIF = '".$nif."'";
    $stmt = odbc_exec($conn, $sql);
    if(odbc_fetch_row($stmt)){
        $idBorrar = odbc_result($stmt, 'id');
    }
    odbc_free_result($stmt);
    $sql1 = "delete Otros_titulos WHERE id_persona = '".$idBorrar."'";
    $sql2 = "delete Otros_datospersonales WHERE id_persona = '".$idBorrar."'";
    $sql3 = "delete Otros_datos WHERE id_persona = '".$idBorrar."'";
    $sql4 = "delete Otros_cursos WHERE id_persona = '".$idBorrar."'";
    $sql5 = "delete Idiomas WHERE id_persona = '".$idBorrar."'";
    $sql6 = "delete Futuro_empleo WHERE id_persona = '".$idBorrar."'";
    $sql7 = "delete Estudios WHERE id_persona = '".$idBorrar."'";
    $sql8 = "delete Conocimientos_tecnologias WHERE id_persona = '".$idBorrar."'";
    $sql9 = "delete Experiencia WHERE id_persona = '".$idBorrar."'";
    $sql10 = "delete Datos_personales WHERE id = '".$idBorrar."'";

    for($j=1;$j<12;$j++){
        $stmt = odbc_exec($conn, "{$sql".$j}");
        odbc_free_result($stmt);
    }
    $control=$x;
}
```

Ilustración 70: Control de CV repetidos

Para controlar este problema, el programa hace lo siguiente. Cuando se está leyendo el CV y obtenemos el valor del DNI de la persona (en la captura, `nif`), se hace primero una búsqueda en la base de datos. En caso de existir el DNI, entonces se realiza un borrado de la base de datos donde tenga que ver la persona de ese CV, y después, se realiza la inserción normal. Esta decisión ha sido así porque en caso de insertar dos CVS iguales, solo quedará uno en la base de datos. En caso de insertar un CV una vez, y luego volver a insertarlo con modificaciones, pues el anterior CV no tiene validez, por lo que se borra y se inserta el nuevo.

Todo este proceso es la espina fundamental del sistema. Un programa que sea capaz de leer automáticamente cualquier CV en formato html de Infojobs y Tecnoempleo, e insertarlo en la base de datos. A continuación se va a explicar el proceso para el desarrollo de la página web, donde se puede visualizar toda la información y gestionarla.

5.3. Página web

La realización de la página web tiene varios factores que es necesario que se combinen entre ellos para conseguir el resultado esperado. Los factores son los siguientes:

- **Html:** mediante el lenguaje de etiquetas html, se ha creado el diseño de la página. Este diseño tiene que ver con las distintas zonas de la página, como por ejemplo, una cabecera arriba de la página, un cuerpo donde incluir botones, inputs para rellenar como un formulario o como criterios de búsqueda, etc.
- **Css:** mediante hojas de estilo css se ha dado estilo a la página web. El color de cada apartado, el tamaño, la forma...todo ha sido especialmente diseñado mediante las hojas de estilo.
- **JavaScript:** mediante este lenguaje de programación, hemos realizado la funcionalidad de la página. Esta funcionalidad consiste principalmente en los botones o inputs, que redirijan a páginas concretas, que se dupliquen para incluir información, que muestren ventanas emergentes...etc.
- **PHP:** mediante este lenguaje de programación, hemos realizado el manejo de datos que se tienen que mostrar en la página, al mismo tiempo que los datos que se tienen que guardar o modificar en la base de datos. En caso de hacer una búsqueda de un CV y pincharlo para querer ver su información, se mostrarán todos los campos, y estos campos se auto rellenan mediante PHP para que se pueda ver la información. Se realizan búsquedas en la base de datos sobre el identificador de ese CV y después esa información se guarda en variables, que se pasan como valores a los campos de la página. De la misma manera ocurre al revés. Si queremos dar de alta un CV, aparecerá un formulario con todos los campos a rellenar vacíos. Al rellenarlos, recogeremos con PHP todos los valores introducidos en cada campo, y pasaremos a realizar las inserciones o modificaciones a la base de datos.

Por tanto, con html y css se crea una página vistosa y agradable donde se puede ver con claridad el funcionamiento del sistema. Con JavaScript, dotamos a la página de la funcionalidad necesaria, y finalmente con PHP realizamos todo el manejo de datos, tanto para mostrarlos como para guardarlos o modificarlos.

A continuación se va a explicar en detalle la creación de algunas páginas, las más importantes, que sirven como guía para la creación de las demás.

Index

Lo primero que vamos a explicar es como se realiza la comprobación de datos para poder entrar a la aplicación. La pantalla de Index es la siguiente:

Ilustración 71: Pantalla de Index

Para entrar a la aplicación, hay que introducir un correo electrónico y una contraseña. Para ello, ya tienes que estar registrado en la aplicación anteriormente, y por tanto un administrador ha tenido que dar de alta a ese usuario en concreto.

Ilustración 72: Crear usuario

A la hora de registrar al usuario (cosa que solo pueden los administradores), poniendo cualquier contraseña, el programa introducirá al usuario en la base de datos en su tabla "Usuarios", pero la contraseña la introducirá cifrada.

Este cifrado se hace de la siguiente manera:

```
class Encrypter {

    private static $Key = "dublin";

    public static function encrypt ($input) {
        $output = base64_encode(mcrypt_encrypt(MCRYPT_RIJNDael_256, md5(Encrypter::$Key), $input, MCRYPT_MODE_CBC, md5(md5(Encrypter::$Key))));
        return $output;
    }

    public static function decrypt ($input) {
        $output = rtrim(mcrypt_decrypt(MCRYPT_RIJNDael_256, md5(Encrypter::$Key), base64_decode($input), MCRYPT_MODE_CBC, md5(md5(Encrypter::$Key))), "\0");
        return $output;
    }

}
```

Ilustración 73: Método cifrado

Por tanto, en la base de datos tendremos una contraseña que habrá que descifrar al entrar. En la página de Index se rellenarían los datos, poniendo el correo electrónico y la contraseña no cifrada que eligió el usuario. El programa hará lo siguiente:

```
if(isset($_POST['inputUsu']) && isset($_POST['inputPassword'])) {
    if($_POST['inputUsu'] != '' && $_POST['inputPassword'] != '') {
        session_start();
        $usuario = $_POST['inputUsu'];

        $pass = Encrypter::encrypt($_POST['inputPassword']);

        $sql = "SELECT * FROM Usuarios WHERE correo = '". $usuario. "' AND pass = '". $pass. "'";

        $stmcomprobar = odbc_exec($conn, $sql);
        if(odbc_fetch_row($stmcomprobar)) {
            $result = odbc_result($stmcomprobar, 'correo');
            $result2 = odbc_result($stmcomprobar, 'perfil');
            $result3 = odbc_result($stmcomprobar, 'id');
        }
        if($result != null) {
            $_SESSION["usuario"] = $usuario;
            $_SESSION["rol"] = $result2;
            $_SESSION["id_usuario"] = $result3;

            session_write_close();
            header("Location: /InfojobsPRE/Inicio.php");
        }
    }
}
```

Ilustración 74: Comprobar contraseña

Se detectaría la contraseña mediante `$_POST["inputPassword"]`, y se le pasaría el método de cifrado, `Encrypter::encrypt`. De esta manera, estamos cifrando la contraseña introducida, y se hará una comprobación de correo electrónico y contraseña en la tabla Usuarios. Si coinciden, entonces el usuario podrá entrar.

A la hora de crear al usuario, aparte de guardarlo con la contraseña cifrada, se le asignará un perfil entre los siguientes:



Ilustración 75: Perfiles de usuario

Cada perfil tiene unos privilegios específicos. Así por ejemplo, el administrador podrá acceder a toda la funcionalidad, mientras que el perfil Usuario externo sólo podrá realizar búsqueda de CVS, ni dar de alta, ni acceder a peticiones, ni acceder por supuesto al mantenimiento de la aplicación. Este control se realiza de la siguiente manera, por ejemplo para la pantalla de inicio, siendo administrador:



Ilustración 76: Inicio Administrador

Se muestran todos los módulos disponibles. Sin embargo, para un usuario cuyo perfil es Usuario externo:



Ilustración 77: Inicio Usuario externo

Este usuario sólo podrá acceder a la búsqueda de candidatos. Este control se consigue mediante lo siguiente:

```

<input id='importar2' type='button' value="Búsqueda" onclick="funcion2()">
<br>
<br>
<input id='importar3' type='button' value="Dar de alta" onclick="funcion3()">
<br>
<br>
<input id='importar4' type='button' value="Peticiones" onclick="funcion4()">
</div>
</body>
</html>
<script language=Javascript>
$(document).ready(function() {
    if('<?php echo $rol;?>'=='Petición'){
        $("#importar2").hide();
        $("#importar3").hide();
    }else if('<?php echo $rol;?>'=='Técnico'){
        $("#importar3").hide();
    }else if('<?php echo $rol;?>'=='Gestor'){
        $("#importar3").hide();
    }else if('<?php echo $rol;?>'=='Consulta'){
        $("#importar3").hide();
        $("#importar4").hide();
    }else if('<?php echo $rol;?>'=='Usuario externo'){
        $("#importar3").hide();
        $("#importar4").hide();
    }
    if('<?php echo $rol;?>'!='Administrador'){
        $("#mantenimiento").hide();
    }
}

```

Ilustración 78: Ejemplo de control perfiles

Vemos en el html como los botones de "Dar de alta" y de "Peticiones" tienen un identificador que es "importar3" e "importar4". Mediante JavaScript, comprobamos que tipo de rol de usuario es el que está accediendo a la página. En este caso, si es "Usuario externo", se les aplicará la clase "hide" a los botones "importar3" e "importar4", por lo que no aparecerán para ese usuario y se ocultarán. Además, más abajo se comprueba que si el rol del usuario es distinto a "Administrador", siempre se ocultará el botón de mantenimiento. Con esto tendríamos controlado prácticamente todo el acceso a cada tipo de usuario. Sin embargo, falta una cosa.

En caso de que alguien escribiera en la barra de direcciones una dirección válida pero a la cual no tuviera permitido el acceso, aún así podría acceder. No basta con ocultar los botones para acceder a las páginas, hay que hacer algo más. Por ejemplo, si un usuario cuyo perfil no es "Administrador" y es cualquier otro, y escribe:

"http://localhost/InfojobsPRE/Mantenimiento.php"

Entraría en la página de mantenimiento, y eso no es posible, Por eso, en todas las páginas, al inicio se comprueba que hay una sesión iniciada, es decir, que un usuario ha entrado a la aplicación aportando sus datos, y por tanto se realiza una comprobación.

En este caso, si el usuario que está en la página "Mantenimiento.php" es distinto de "Administrador", automáticamente se redirige a la página de inicio.

```
<?php
session_start();
if($_SESSION["usuario"]==""){
    header("Location:/InfojobsPRE/Index.php");
}
if($_SESSION["rol"]==""){
    header("Location:/InfojobsPRE/Index.php");
}else{
    $rol = $_SESSION["rol"];

    if($rol!='Administrador'){
        header("Location:/InfojobsPRE/Inicio.php");
    }
}
```

Ilustración 79: Comprobar perfil en Mantenimiento.php

Cabeceras

En primer lugar, tenemos la cabecera de todas las páginas, un ejemplo:



Ilustración 80: Cabecera de páginas

El código html es el siguiente:

```
<body>


<header class="Cabecera">

    <h3 style="color:white">SELECCIÓN DE CV</h3>
    <div class="botones-cabecera">
        <button class="atras" id='cerrar' type="button" style='float:right' onclick="cerrarSesion()">Cerrar sesión <i class="glyphicon glyphicon-log-out">
    </div>
</header>
<fieldset class="subHeader" >
    <button id='mantenimiento' type="button" style='float:right' onclick="funcion5()">Mantenimiento <i class="glyphicon glyphicon-cog"></i></button>
</fieldset>
```

Ilustración 81: Html de cabecera de páginas

Vemos como al principio disponemos de dos imágenes que sirven como fondo de la aplicación. Después tenemos el título de la página que en este caso es "SELECCIÓN DE CV", seguido de un botón para cerrar sesión. Justo debajo, tenemos otro botón, en este caso, de mantenimiento. Vamos a ver ahora el código JavaScript correspondiente:

```
function funcion5(){
    location.href='/InfojobsPRE/Mantenimiento.php';
}
function cerrarSesion(){
    location.href='/InfojobsPRE/CerrarSesion.php';
}
```

Ilustración 82: JavaScript de cabecera de páginas

El botón de cerrar sesión tiene asignada la función cerrarSesion() que se encarga de redirigir al archivo CerrarSesion.php que destruye la sesión abierta y redirige a la página de index. Por otro lado, el botón mantenimiento tiene asignado la función funcion5, que se encarga de redirigir la página al archivo Mantenimiento.php.

Alta candidato

Una vez analizada la cabecera, vamos a analizar el cuerpo de la página utilizando por ejemplo la página de alta de un candidato. La creación de esta página supone la realización de un formulario específico, que una vez relleno, tiene que existir una opción de poder guardarlo. La página resultante es la siguiente:

Ilustración 83: Pantalla de alta candidato

Como vemos, la cabecera ha cambiado respecto a lo explicado antes, pero la estructura es la misma. El título ha cambiado de "SELECCIÓN DE CV" a "NUEVO CV" ya que tenemos que indicar que esta página se encarga de crear un nuevo CV. El botón de cerrar sesión sigue, y además se han añadido dos botones que permiten volver atrás, o volver a la página de inicio. En lugar de mantenimiento, ahora tenemos dos botones que realizan otras funciones: uno permite guardar el CV una vez relleno el formulario, y el otro redirige a la página de importar un CV existente, es decir, importar un CV en formato html. Como vemos, las cabeceras de las páginas siempre tendrán la misma estructura, solo que tendrán pequeñas modificaciones dependiendo de la página en concreto.

Una vez analizadas las cabeceras y sus posibles cambios, vamos a centrarnos en el cuerpo. En este caso, tenemos un formulario, y vamos a ver el código:

```
<form id="CV" name="CV" action="" onsubmit="return validateForm()" method="post">
<div id="datosalta" class="datos">
<h4 class="titulo glyphicon glyphicon-chevron-up">DATOS PERSONALES</h4>
<div id="datospersonales">
<div>
<label>Nombre completo</label>
<input type="text" id="nombreCompleto" name="nombreCompleto" placeholder="Nombre completo" value="">
<b>*</b>
</div>
<div>
<label>E-mail</label>
<input type="text" id="E-mail" name="E-mail" placeholder="E-mail" value="" onblur="comprobarEmail()">
<b>*</b>
</div>
<div>
<label>Sexo</label>
<div id="dSexo">
<input type="radio" id="sexo" class="inIzq" name="sexo" value="Hombre">Hombre
<input type="radio" id="sexo" class="inDer" name="sexo" value="Mujer">Mujer
</div>
<b>*</b>
</div>
</div>
```

Ilustración 84: Html alta candidato

Mediante etiquetas h4 se crean las cabeceras de cada sección, en este caso de datos personales. La estructura es la siguiente:

Se crea un div que contendrá un label, que será una etiqueta de lo que representa el campo, y luego un input donde poder escribir el valor. Por ejemplo, el primer label tiene como valor "Nombre completo" y el input es de tipo "text", dejando posibilidad de escribir lo que sea. Después irá el div del email, luego del sexo, luego de fecha de nacimiento...y así hasta finalizar. Cabe mencionar, que para las provincias, se ha creado un select que tendrá los valores de la tabla provincia explicado en el apartado 5.1 Base de datos, o que sexo es de tipo check, pudiendo seleccionar "Hombre" o "Mujer". Vamos a ver ahora un ejemplo de código CSS que complementa a esta página:

```

[ #datosalta #datospersonales label {
    margin-bottom: 15px;
    margin-right: 10px;
    text-align: right;
    width: 120px;
- }
[ #datosalta #datospersonales input[type="text"] {
    width: 350px;
- }

```

Ilustración 85: CSS de alta de candidato

Aquí podemos ver cómo le damos un estilo a los label contenidos dentro de datos personales. Todos tienen márgenes específicos, con el texto alineado a la derecha, y con un ancho específico. Así mismo, los input tipo "text", tienen un ancho de 350px. Todo esto ha sido colocado de tal manera que quedara lo más vistoso posible.

A continuación se va a explicar la funcionalidad de los botones que añaden campos nuevos. En la captura de alta de candidato, podemos ver como hay un botón de "Añadir Contacto 1", y debajo en el apartado de estudios vemos un botón de "Quitar Estudio 1" y otro de "Añadir Estudio 2". Esto es porque se ha pinchado en un botón que era "Añadir Estudio 1". Este botón ha producido que aparezca la etiqueta de Estudio 1 y todos los inputs posteriores, para poder rellenar un estudio de una persona. Además, ha cambiado el botón de "Añadir Estudio 1" por "Quitar Estudio 1" y ha añadido un botón de "Añadir Estudio 2".

Esto es por lo siguiente. Inicialmente, hay un div creado con el valor del label "Estudio 0" y los 4 inputs siguientes, todo ello oculto mediante CSS (añadiendo la clase "hide"). Sin embargo, el botón de "Añadir Estudio 1" se muestra. Al pinchar ahí, se ejecuta una función llamada addEstudio() de JavaScript, que lo que hace es clonar el div oculto, y le quita la clase "hide", por lo que se muestra, y además renombra los botones. De esta manera, de tener un div oculto con un botón de "Añadir Estudio 1", pasamos a tener otro div idéntico, pero sin estar oculto, del botón "Añadir Estudio 1" pasamos a tener "Quitar Estudio 1" y el botón clonado se renombra a "Añadir Estudio 2".

A continuación vemos el código html:

```
<h4 class="titulo glyphicon glyphicon-chevron-up">ESTUDIOS</h4>
<div id="estudios">
  <div id="divestudios0" class="estudios">
    <div>
      <label id='labelEstudio0' class="hide">Estudio 0</label>
      <input type="text" id="periodo0" name="periodo0" placeholder="Fecha inicio" readonly>
      <input type="text" id="periodof0" name="periodof0" placeholder="Fecha fin" readonly>

      <?php
        $sql = "SELECT titulacion FROM Titulaciones ORDER BY titulacion";
        $stmcomprobar = odbc_exec($conn, $sql);
        $titulaciones=array();

        while(odbc_fetch_row($stmcomprobar)){
          $result = utf8_encode(odbc_result($stmcomprobar,'titulacion'));
          array_push($titulaciones, $result);
        }

        echo "<select id=\"titulo0\" name=\"titulo0\" >";

        echo "<option value=\"nada\">";
        echo ">Seleccionar titulación</option>";

        foreach ($titulaciones as $k => $v) {
          echo "<option value=\"\$v\">";
          echo ">$v</option>";
        }
        echo "</select>";
      ?>
      <input type="text" id="especialidad0" name="especialidad0" placeholder="Especialidad">
      <input type="text" id="centro0" name="centro0" placeholder="Centro">
    </div>
  </div>
</div>
```

Ilustración 86: Html Estudios

Aquí podemos ver como todos los nombres están acompañados de un 0. Esto lo ocultamos con javascript:

```
function ocultarEstudios(){
  if($("#titulo0") && $("#especialidad0") && $("#centro0") && $("#periodo0") && ($("#tituloEstudios0")){
    $("#tituloEstudios0").hide();
    $("#titulo0").hide();
    $("#especialidad0").hide();
    $("#centro0").hide();
    $("#periodo0").hide();
    $("#periodof0").hide();
  }
}
```

Ilustración 87: Funcion ocultarEstudios()

Forzamos a que los nombres que tengan un 0 tengan la clase "hide" y por tanto aparezcan ocultos. Haríamos esto con todos menos con el botón de añadir, que quedaría al descubierto:

```
<div style="text-align:center" class="boton-masestudio">
  <input id='anadir0' class="masestudio" type='button' value="Añadir Estudio 1">
</div>
```

Ilustración 88: Botón añadir Estudios

Este botón tiene añadida la clase "masestudio", que mediante javascript le añadimos una función específica:

```
$(".masestudio").each(function (el){
  $(this).bind("click",addEstudio);
});
```

Ilustración 89: Clase masestudio

Le estamos diciendo que al pinchar en el botón "Añadir Estudio 1" se ejecute la función `addEstudio`. Un ejemplo de la funcionalidad de esta función:

```
if(estudioID==0&&$("#titulo"+nuevoEstudioID)&&$("#especialidad"+nuevoEstudioID)&&$("#centro"+nuevoEstudioID)
&&$("#periodo"+nuevoEstudioID)&&$("#periodof"+nuevoEstudioID)){
$("#anadir"+estudioID).val('Quitar Estudio 1').unbind("click",addEstudio);
$("#anadir"+estudioID).bind("click",borrarEstudio);
$("#anadir"+nuevoEstudioID).val('Añadir Estudio 2');
$("#titulo"+nuevoEstudioID).show();
$("#especialidad"+nuevoEstudioID).show();
$("#centro"+nuevoEstudioID).show();
$("#periodo"+nuevoEstudioID).show();
$("#periodo"+nuevoEstudioID).attr("class","");
```

Ilustración 90: Función `addEstudio()`

Esta función ha calculado dos identificadores: `estudioID` es el número que acompaña al nombre cuando se pincha en el botón. Es decir, al pinchar por primera vez, el input correspondiente al centro donde ha estudiado, tiene de nombre "centro0". Pues bien, `estudioID` valdrá 0, y `nuevoEstudioID` valdrá `estudioID+1`, en este caso 1. Por tanto, para el input `centro1` le aplica un ".show", es decir, lo muestra, y entonces pasa de estar oculto a mostrarse. Así lo hace con todos los inputs, además de al botón inicial, cambiar su nombre de "Añadir Estudio 1" a "Quitar Estudio 1". Además, la función `unbind` hace que ese botón ya no tenga asociado al pinchar la función "addEstudio", y con la función `bind` le añadimos la función "borrarEstudio", ya que ahora podremos eliminar el estudio que hemos creado.

La función "borrarEstudios" se encarga de borrar el div que contiene toda la información de los estudios, y renombrar los botones, es decir, que si hay dos estudios y borramos el primero, el segundo estudio tiene que pasar a ser el primero. Un ejemplo de la funcionalidad de esta función es:

```
function borrarEstudio() {
//variable que contiene el numero del div que borramos
var borrar = parseInt($("#this").parent('div').parent('div').attr('id').replace('divestudios',''));

var contador = borrar +1;

while (contador>=1){
if(borrar==0&&document.getElementById("divestudios"+contador)){
$("#divestudios"+contador).remove();
```

Ilustración 91: Función `borrarEstudio()`

Lo que hace esta función es detectar el div que queremos borrar recogiendo el número final del nombre del div, y dependiendo de eso, borra el div con la función `remove()`.

Ya se ha comentado como cambia la cabecera, como se muestran las entradas y etiquetas en pantalla, y cómo funcionan los botones para duplicar y poder añadir toda la información que se desee. Lo último que queda, es guardar estos cambios de manera que se pueda dar de alta a un candidato.

Arriba a la derecha, tenemos un botón que indica "Guardar cambios". Al pinchar en él, se muestra un div oculto, de tal manera que parezca un "pop-up". Lo podemos ver a continuación:

Ilustración 92: Guardar cambios alta candidato

Una vez mostrado el "pop-up", tenemos un mensaje de confirmación, y dos botones.

```
<div class="overlay-container">
  <div class="window-container zoomin">
    <h4>¿Estás seguro de guardar los cambios?</h4>
    <input class="close" type='button' value="Cancelar">
    <input class="close" name="guardar" type='submit' value="Aceptar" form="CV">
  </div>
</div>
```

Ilustración 93: Pop-up de guardar cambios

El botón de "Cancelar" simplemente vuelve a ocultar el "pop-up". Sin embargo, el botón de aceptar, realiza un submit del form "CV". Esto es lo siguiente. Todo el formulario de alta de candidato, está dentro de una etiqueta form:

```
<form id="CV" name="CV" action="" onsubmit="return validateForm()" method="post">
  <div id="datosalta" class="datos">
    <h4 class="titulo glyphicon glyphicon-chevron-up">DATOS PERSONALES</h4>
    <div id="datospersonales">
```

Ilustración 94: Form alta candidato

Este form se llama "CV", y la acción que tiene definida es vacía. Esto hace que se recargue la página. Por tanto, al pinchar en el botón de aceptar, se recargará la página en la que estamos. Al recargarse la página, ocurre lo siguiente, y es que se envían los datos del formulario, así como la información de los botones, digamos por detrás, sin que se vea. Estos datos los podemos utilizar de la siguiente manera. Si rellenamos los datos personales y le damos a guardar cambios y luego a aceptar, esos datos de datos personales se envían por detrás, y al recargarse la página podemos utilizarlos así:

`$_POST['nombreCompleto']`

La variable POST indica que ese valor viene de una página anterior (al recargar estamos en una página nueva), y 'nombreCompleto' hace referencia al nombre de un input concreto. De esta manera, obtenemos el valor del nombre que había escrito en el formulario. Por tanto,

podremos recoger todos los valores de todos los inputs que se hayan rellenado, y realizar una inserción en la base de datos. Un ejemplo:

```

$i=1;

while($i<$j){
    if($_POST['fecha_obtencion'][$i]!=''||$_POST['nombre_titulo'][$i]!=''||$_POST['centro_titulo'][$i]!=''){
        $id4 = NewGuid();
        $sql_otrostitulos = "INSERT INTO Otros_titulos(id,id_persona, fecha_obtencion, nombre_titulo, centro)
        VALUES ('".$id4."','".$id."','".$_POST['fecha_obtencion'][$i]."', '".$_POST['nombre_titulo'][$i]."', '".$_POST['centro_titulo'][$i]."'");
        $odbc_exec($conn, $sql_otrostitulos);
    }
    $i++;
}

```

Ilustración 95: Inserción de Otros_titulos en alta candidato

Se crea un bucle que recorra todas las variables POST que tengan valor para los nombres fecha_obtencion, nombre_titulo y centro_titulo. Por cada iteración del bucle, tendremos tres valores para cada variable, y por tanto realizamos la inserción a la base de datos. Este proceso se repite para todo el formulario.

En resumen, al pinchar en guardar cambios y aceptar, se comprueban todas las variables POST que tienen valor, y una vez obtenidos los valores, se realizan las inserciones en la base de datos. Finalmente, se muestra un mensaje de confirmación:

The screenshot shows a web form titled "NUEVO CV". At the top right, there are buttons for "Atrás", "Inicio", and "Cerrar sesión". Below these are "Importar CV existente" and "Guardar cambios". A message box states: "La inserción de Sergio se ha realizado correctamente". The form is divided into sections: "DATOS PERSONALES", "ESTUDIOS", and "OTROS TÍTULOS". The "DATOS PERSONALES" section includes fields for "Nombre completo", "E-mail", "Sexo" (radio buttons for Hombre and Mujer), "Fecha de nacimiento", "Dirección", "Código postal", "Población", "Provincia" (dropdown), "NIF/NIE", "Teléfono fijo", "Teléfono móvil", and "Teléfono extranjero". There is an "Añadir Contacto 1" button at the bottom of this section. The "ESTUDIOS" section has an "Añadir Estudio 1" button. The "OTROS TÍTULOS" section is currently empty.

Ilustración 96: Inserción correcta de alta candidato

La explicación de esta página sirve también para dar de alta de peticiones, pues es prácticamente igual. La única diferencia es que las peticiones tienen un "estado". Éste se inicializa automáticamente a "nueva" al crear la petición, pasará a "abierta" cuando se esté editando una petición, y pasará a "cerrada" cuando el usuario lo indique a propósito. A continuación tenemos la búsqueda de candidatos.

Búsqueda candidato

La página de búsqueda de un candidato es la siguiente:

Esta página no cambia prácticamente respecto a la anterior, aunque parezca que sí. Al igual que antes, la cabecera cambian muy pocas cosas, el título y que no hay ningún botón en la franja gris, lo demás es idéntico.

En el cuerpo de la página, tenemos una serie de filtros de búsqueda, que no son más que una serie de etiquetas para indicar qué se debe introducir dentro del input. La gracia de esto es la concatenación de todas las entradas para realizar una consulta a la base de datos de tal manera que muestre los resultados obtenidos. Por ejemplo, vamos a buscar a la persona que hemos introducido antes:

Ilustración 97: Resultado búsqueda candidato

Vemos como hemos introducido el nombre y nos ha dado como resultado un valor. Ahora podemos pinchar en él para ver su información completa. Este resultado se ha mostrado mediante una select que tiene esta estructura:

```
SELECT DISTINCT(dbo.Datos_personales.id), nombre_completo FROM dbo.Datos_personales
WHERE nombre_completo LIKE '%sergio%' ORDER BY nombre_completo ASC;
```

Se produce una búsqueda en la base de datos de todas las tuplas cuyo nombre es "Sergio" y se ordenan para mostrarlos. La explicación de esta página sirve también para búsqueda de peticiones, pues es prácticamente igual.

La creación de la página web se basa en estos conceptos que se han explicado. Se dispone de código html para mostrar el apartado visual, y mediante CSS se maqueta y se le da un estilo propio. Con JavaScript damos la funcionalidad concreta y con PHP accedemos a la base de datos. Las demás páginas de la aplicación son variaciones de las ya explicadas, ya que por ejemplo dar de alta una petición es igual que dar de alta un candidato, mostrar la información de la base de datos se hace mediante PHP y se incluye en los inputs necesarios, etc. El desarrollo concluye con la creación de la página web. Una vez tenemos la página web visible y en pleno funcionamiento, incluimos un apartado en ella que sea la de importar un CV. Esta sección se compone de un seleccionador de archivos donde podremos seleccionar todos los CVS que se deseen, y mediante un botón de enviar, se cargarán todos los CVS a la base de datos. Simplemente hay que añadir nuestro programa principal de importación de CVS a la página web en un apartado especial. En el siguiente apartado, en el de "Resultado", se mostrarán todas las páginas resultantes y su funcionalidad.

6. Resultado

En este apartado se va a mostrar y a explicar el resultado final obtenido con el SGCV. La principal funcionalidad que tiene que ofrecer es la de poder importar un CV automáticamente para poder tratarlo después. No obstante, se va a mostrar también toda la función extra, con capturas de pantalla y con comentarios de cada paso.

Primero vemos como acceder a la aplicación:



Ilustración 98: Resultado Index

Al introducir mis datos, paso a la página de inicio:



Ilustración 99: Resultado Inicio

Como el que está entrando a la aplicación es un usuario administrador, tiene acceso a todo. En primer lugar vamos a dar de alta un usuario pinchando en "Dar de alta":

Ilustración 100: Resultado Alta

Voy a rellenar manualmente el formulario, y a darle a guardar cambios.

The screenshot shows the 'NUEVO CV' (New CV) form. At the top, there are buttons for 'Atrás', 'Inicio', and 'Cerrar sesión'. Below these are 'Importar CV existente' and 'Guardar cambios'. The form is divided into sections: 'DATOS PERSONALES', 'ESTUDIOS', and 'OTROS TÍTULOS'. In the 'DATOS PERSONALES' section, fields for 'Nombre completo', 'E-mail', 'Sexo', 'Fecha de nacimiento', 'Dirección', 'Código postal', 'Población', and 'Teléfono fijo' are visible. A modal dialog box is open in the center, asking '¿Estás seguro de guardar los cambios?' with 'Aceptar' and 'Cancelar' buttons.

Ilustración 101: Resultado prueba de alta candidato formulario

El resultado es:

The screenshot shows the 'NUEVO CV' form after successful submission. A message box at the top states 'La inserción de Sonia se ha realizado correctamente'. The form fields are now empty, with placeholder text like 'Nombre completo', 'E-mail', 'Sexo', 'Fecha de nacimiento', 'Dirección', 'Código postal', 'Población', 'Provincia', 'NIF/NIE', 'Teléfono fijo', 'Teléfono móvil', and 'Teléfono extranjero'. There are buttons for 'Añadir Contacto 1' and 'Añadir Estudio 1'.

Ilustración 102: Resultado alta de candidato formulario

Ahora vamos a importar un candidato automáticamente. Seleccionamos el botón de "Importar CV existente".

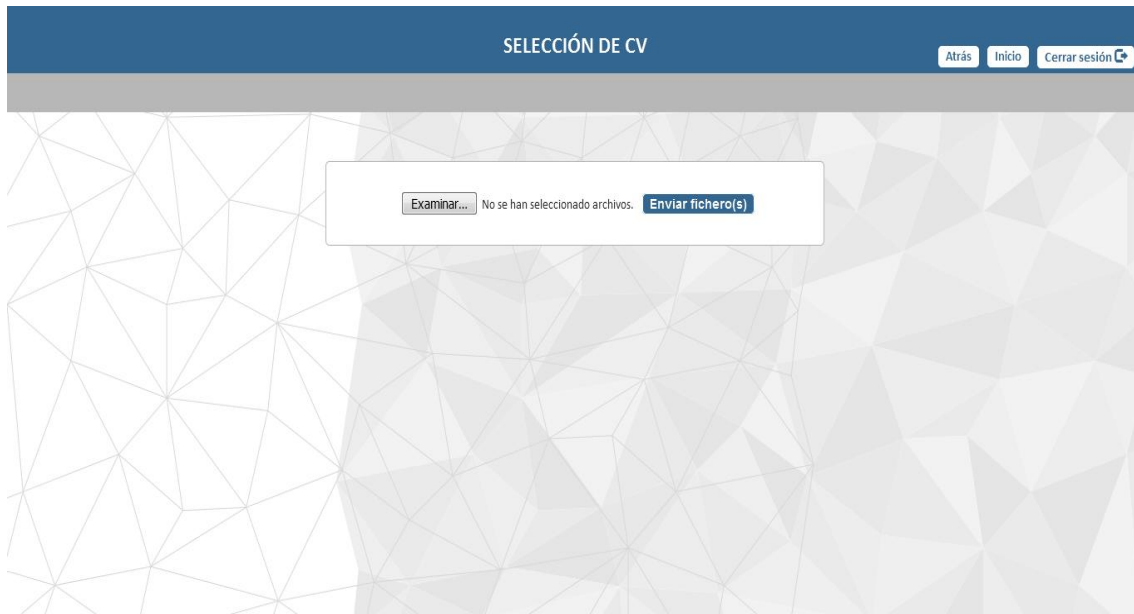


Ilustración 103: Resultado Importar CV-1

Seleccionamos un CV concreto:

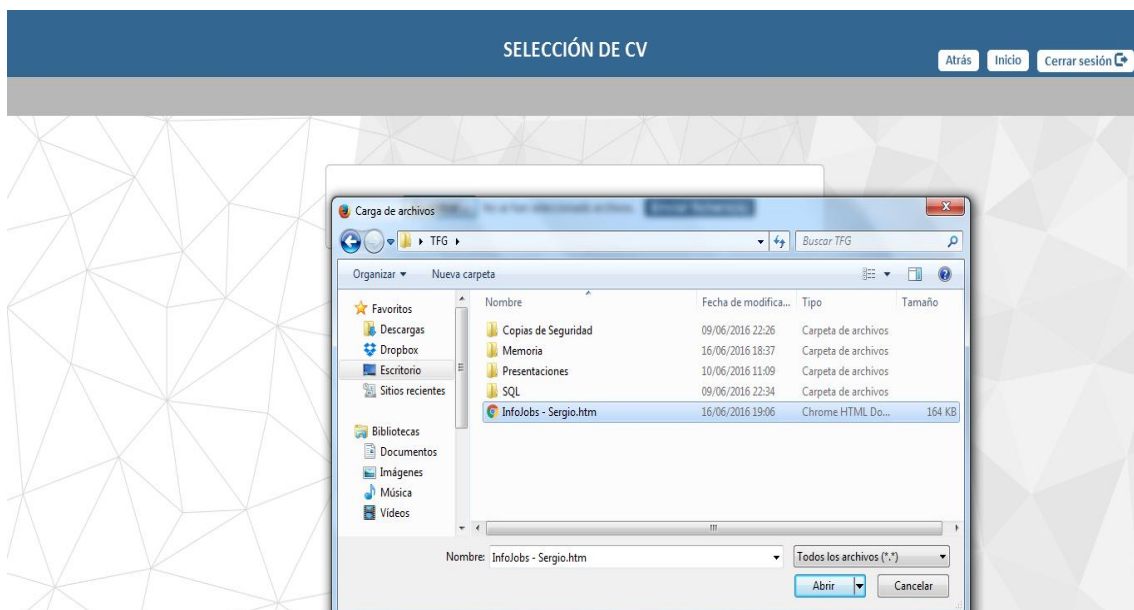


Ilustración 104: Resultado Importar CV-2

Al darle a enviar fichero, el resultado es:



Ilustración 105: Resultado Importar CV-3

Con esto queda explicado el resultado de cómo se importan los CVS. O bien automáticamente o bien manualmente. Ahora vamos a realizar una búsqueda, volviendo a la página de inicio y pinchando en "Búsqueda":

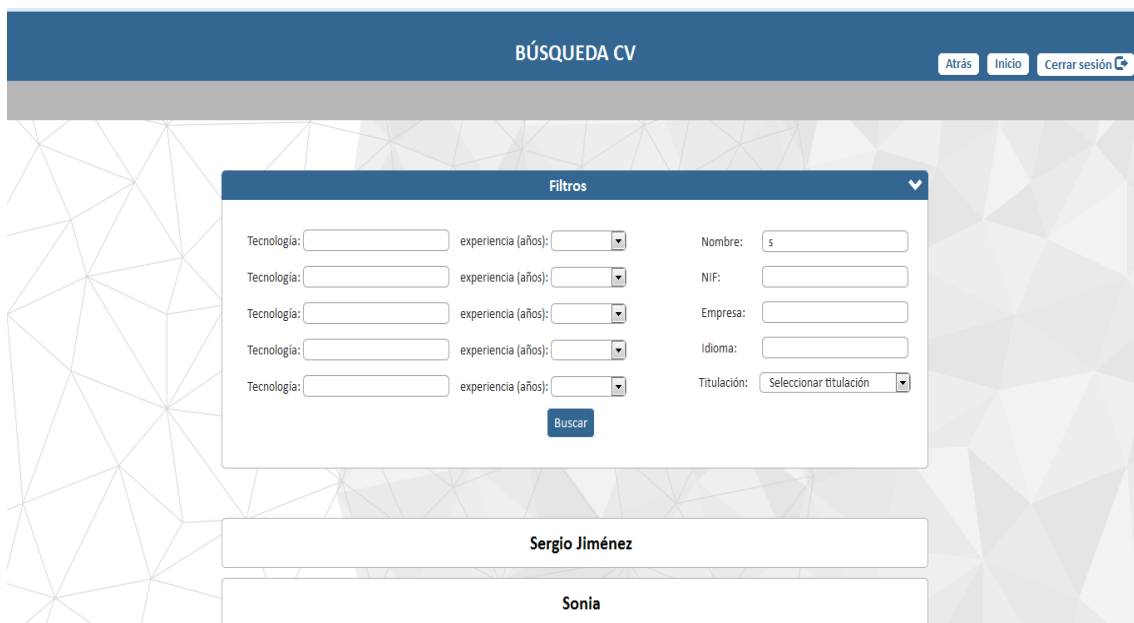


Ilustración 106: Resultado búsqueda-1

Vemos como el resultado de buscar por nombre algo que contenga "s" son los dos CVS que hemos introducido.

Otro ejemplo de búsqueda sería introducir una tecnología que ambos compartan, en este caso "html".

Ilustración 107: Resultado búsqueda-2

Vemos como el resultado es el mismo. Vamos a pinchar por ejemplo en "Sergio Jiménez". A continuación saldría la siguiente página:

Ilustración 108: Resultado mostrarCurrículum

Esta página sirve para mostrar de forma clara la información del CV de la persona en cuestión, y además permite dos opciones. Exportar el CV a WORD en un formato concreto, o poder editar los datos del CV. En primer lugar, probaremos la exportación del CV, pinchando en el botón "Exportar a WORD".

El resultado es el siguiente:

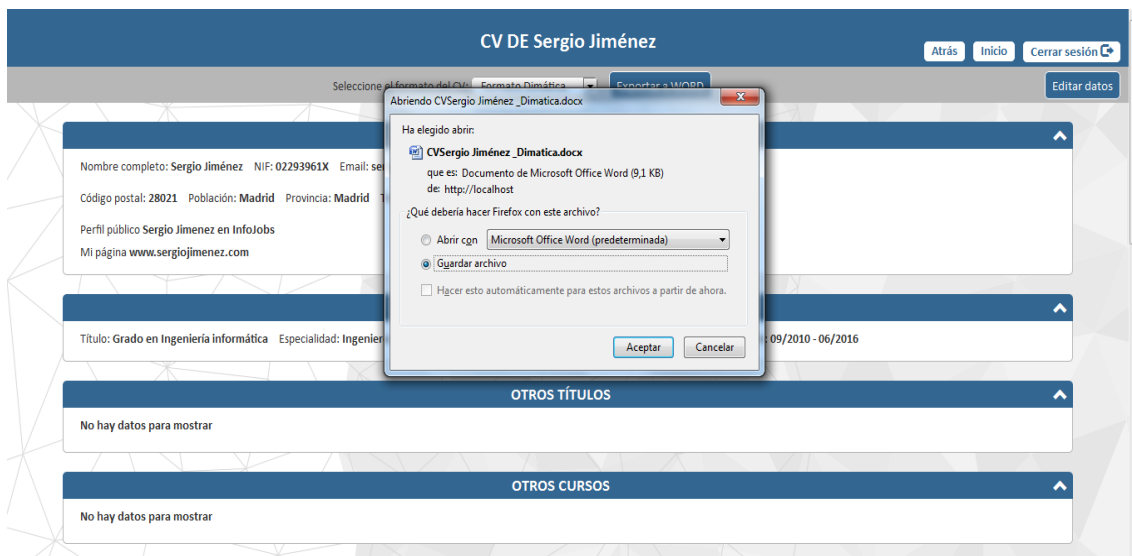


Ilustración 109: Resultado exportar a word-1

Podemos elegir entre abrir el archivo o guardarlo. Una vez abierto, tenemos lo siguiente:

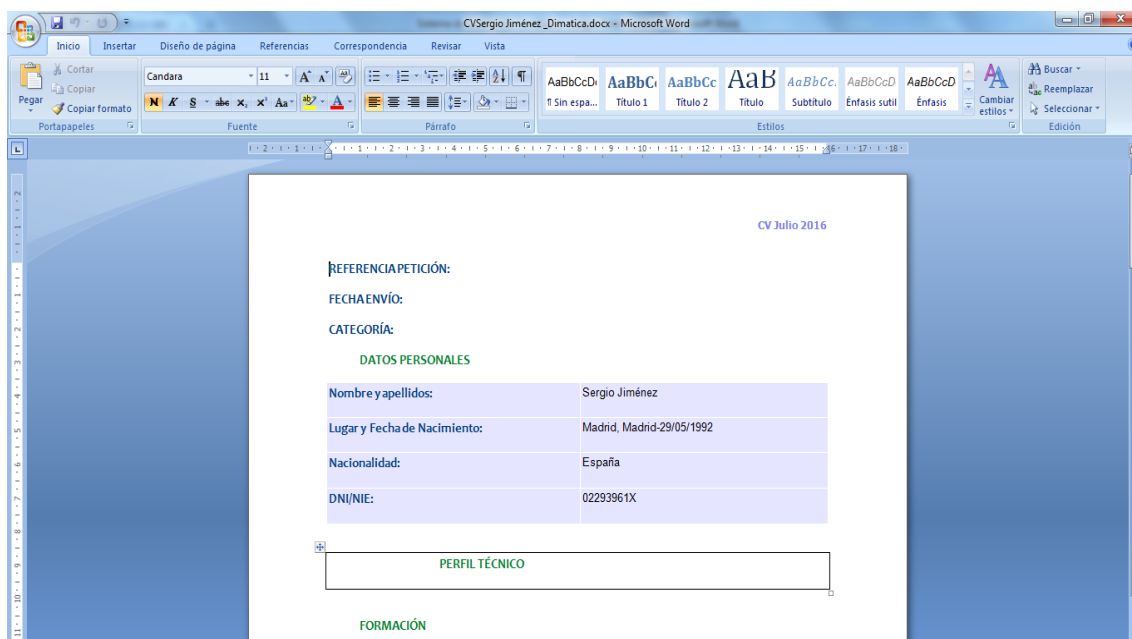


Ilustración 110: Resultado exportar a word-2

El resultado es un Word específico que se auto rellena con los datos del CV. Hay campos vacíos debido a que son campos que no están incluidos en el CV. No obstante, el rellenar los demás facilita y agiliza mucho el proceso. A continuación vamos a editar el CV pinchando en el botón de "Editar datos".

El resultado es el siguiente:

The screenshot shows a web form titled "EDICIÓN DEL CV DE Sergio Jiménez". At the top right, there are buttons for "Atrás", "Inicio", "Cerrar sesión", "Borrar datos", and "Guardar cambios". The form is divided into sections. The "DATOS PERSONALES" section contains the following fields:

- Nombre completo: Sergio Jiménez
- Fecha de nacimiento: 29/05/1992
- Sexo: ☒ Hombre, ☐ Mujer
- E-mail: sergio.sejo@gmail.com
- Dirección: Madrid
- Código postal: 28021
- Población: Madrid
- Provincia: Madrid
- NIF/NIE: 02293961X
- Teléfono fijo: Teléfono Fijo
- Teléfono móvil: 686506054
- Teléfono extranjero: Teléfono Extranjero

Below these fields are buttons for "Quitar Contacto 1", "Quitar Contacto 2", and "Añadir Contacto 3". There are also two contact entries:

- Contacto 1: Perfil público, Sergio Jimenez en infolobs
- Contacto 2: Mi página, www.sergiojimenez.com

The "ESTUDIOS" section is partially visible at the bottom, showing a button for "Quitar Estudio 1".

Ilustración 111: Resultado Edición CV

Esta pantalla es similar a la de mostrarCurrículum, sin embargo, los datos en lugar de sólo mostrarse, se pueden editar. Tenemos la opción del botón "Guardar cambios", y vamos a hacer la prueba cambiando el nombre de "Sergio Jiménez" por el de "Juan". El resultado es el siguiente:

This screenshot shows the same form as before, but with the name changed to "JUAN". A confirmation dialog box is displayed in the center of the screen with the text "¿Estás seguro de guardar los cambios?" and two buttons: "Aceptar" and "Cancelar". The background form fields are dimmed.

Ilustración 112: Resultado Edición Guardar cambios-1

Al pinchar en "Aceptar":

The screenshot shows a web interface for editing a CV. At the top, a blue header bar contains the text "EDICIÓN DEL CV DE JUAN" and navigation buttons: "Atrás", "Inicio", and "Cerrar sesión". Below the header, a grey bar contains "Borrar datos" and "Guardar cambios". A white box with a blue border displays the message: "La modificación del CV de JUAN se ha realizado correctamente". Below this is a form titled "DATOS PERSONALES". The form contains several input fields: "Nombre completo" (JUAN), "Fecha de nacimiento" (29/05/1992), "Sexo" (radio buttons for Hombre and Mujer, with Mujer selected), "E-mail" (sergio.sejo@gmail.com), "Dirección" (Madrid), "Código postal" (28021), "Población" (Madrid), "Provincia" (Madrid), "NIF/NIE" (02293961X), "Teléfono fijo" (Teléfono Fijo), "Teléfono móvil" (686506054), and "Teléfono extranjero" (Teléfono Extranjero). Below these fields are three contact entries: "Contacto 1" (Mi página, www.sergiojimenez.com), "Contacto 2" (Perfil público, Sergio Jimenez en Infojobs), and "Añadir Contacto 3".

Ilustración 113: Resultado Edición Guardar cambios-2

Vemos como sale un mensaje de confirmación y de que el nombre completo ahora es "JUAN". Rápidamente hacemos la comprobación, y volvemos a buscar por nombre "s":

The screenshot shows a web interface for searching CVs. At the top, a blue header bar contains the text "BÚSQUEDA CV" and navigation buttons: "Atrás", "Inicio", and "Cerrar sesión". Below the header, a grey bar contains "Borrar datos" and "Guardar cambios". A white box with a blue border displays the message: "La modificación del CV de JUAN se ha realizado correctamente". Below this is a form titled "Filtros". The form contains several input fields: "Tecnología:" (five empty fields), "experiencia (años):" (five dropdown menus), "Nombre:" (s), "NIF:" (empty), "Empresa:" (empty), "Idioma:" (empty), and "Titulación:" (Seleccionar titulación). Below these fields is a "Buscar" button. Below the search filters is a white box with the name "Sonia".

Ilustración 114: Resultado comprobación edición-1

Vemos como ahora "Sergio Jiménez" no sale como resultado.

Vamos a buscar Juan:

Ilustración 115: Resultado comprobación edición-2

Observamos como Juan sale ahora como resultado. Vamos a seleccionarlo y a volver a la edición, para pinchar en el botón de "Borrar datos". Lo que ocurre es lo siguiente:

Ilustración 116: Resultado borrar candidato-1

Nos sale un "pop-up", que al pinchar en aceptar:



Ilustración 117: Resultado borrar candidato-2

Se nos muestra un mensaje de que se ha ejecutado el borrado correctamente. Vamos a comprobarlo buscando de nuevo a "JUAN".



Ilustración 118: Resultado confirmar borrar candidato

Efectivamente, se ha eliminado al candidato correctamente. A continuación, vamos a volver a Inicio y pinchar en el botón "Peticiónes".

Lo que aparece es lo siguiente:



Ilustración 119: Resultado peticiones

Podemos dar de alta peticiones y buscarlas. Pinchando en "Alta de peticiones", sale lo siguiente:

Ilustración 120: Resultado Alta petición

De la misma manera que para un candidato, se rellenan los campos y se crea una nueva petición.

Ahora vamos a buscar una petición pinchando en "Búsqueda de peticiones". Sale lo siguiente al buscar por estado "nueva":

The screenshot shows a web interface titled "BÚSQUEDA PETICIONES". At the top right, there are links for "Atrás", "Inicio", and "Cerrar sesión". Below the header is a "Filtros" (Filters) panel with the following fields:

- Fecha: (text input)
- Estado: (dropdown menu showing "Nueva")
- Nombre petición: (text input)
- Nombre solicitante: (text input)
- Cliente: (text input)
- Detalle: (text input)

Below the filters is a "Buscar" button. At the bottom of the panel, a result card is displayed: "Candidatos que sepan HTML - 14/06/2016 - nueva", with a "Gestión de petición" button next to it.

Ilustración 121: Resultado búsqueda peticiones

Vemos como hay un resultado de una petición: "Candidatos que sepan HTML". Si pinchamos en ella, veremos su información como la del candidato:

The screenshot shows the details of a petition titled "Candidatos que sepan HTML". At the top right, there are links for "Atrás", "Inicio", and "Cerrar sesión", along with an "Editar datos" button. The details are organized into four sections:

- PETICIÓN**:
 - Nombre petición: Candidatos que sepan HTML
 - Solicitante: Sergio
 - Detalles: HTML
 - Fecha: 14/06/2016
- DESTINO**:
 - Empresa: MIEMPRESA
 - Lugar: MADRID
- CONOCIMIENTOS**:
 - Conocimiento Html: Experiencia 2 — Requerido
 - Conocimiento Html: Experiencia 3 — Deseable
- TITULACIÓN**:
 - Titulación requerida: Grado
 - Titulación deseable: Máster

Ilustración 122: Resultado mostrar petición

También hay un botón de editar datos, que funciona exactamente igual que para un candidato. Se podrá borrar la petición o modificar su información. Ahora vamos a lo importante de las peticiones.

En la captura anterior, al tener el resultado de una búsqueda, tenemos un botón que es "Gestión de petición". Al pinchar en él:



Ilustración 123: Resultado gestión peticiones-1

Esta página sirve para poder agregar CVS a la petición, y posteriormente seguimientos sobre esos CV. Primero vamos a agregar un CV a la petición, pinchando en "Agregar CV".



Ilustración 124: Resultado agregar CV a petición

Se abre de nuevo los filtros de búsqueda de un candidato, y buscamos la tecnología "html" como indica la petición. Se muestran dos resultados, con un check-box para poder marcarlo. En este caso marcamos los dos, y le damos al botón de "Agregar a petición".

Se nos muestra un "pop-up", y pinchando en "Aceptar":

The screenshot shows the 'GESTIÓN DE PETICIONES' interface. At the top, there are buttons for 'Atrás', 'Inicio', and 'Cerrar sesión', along with a 'Guardar cambios' button. The main content is divided into three sections: 'PETICIÓN', 'SEGUIMIENTO', and 'CV ASOCIADOS'. The 'PETICIÓN' section shows a petition with the name 'Candidatos que sepan HTML' and the status 'abierta', with a 'Ver' button. The 'SEGUIMIENTO' section has an 'Añadir Seguimiento 1' button. The 'CV ASOCIADOS' section lists two CVs: 'Sergio Jiménez' and 'Sonia', both associated with the date '16 Junio 2016'. Each CV entry has 'Ver' and 'Quitar' buttons. At the bottom of the CV list is an 'Agregar CV' button.

Ilustración 125: Resultado gestión peticiones-2

Automáticamente se vuelve a mostrar la página de gestión de peticiones, pero esta vez aparecerán los dos CVS asociados a la petición. Pinchando en el botón de "Ver" podremos ver el CV sin cambiar de página, y además exportarlo a Word:

The screenshot shows a pop-up window titled 'verCV asociado'. At the top, there is a dropdown menu for 'Seleccione el formato del CV:' with 'Formato Dimática' selected, and an 'Exportar a WORD' button. Below this is a section titled 'DATOS PERSONALES' which contains the following information: 'Nombre completo: Sergio Jiménez', 'Email: sergio.sejo@gmail.com', 'Sexo: Hombre', 'Fecha de nacimiento: 29/05/1992', 'Código postal: 28021', 'Población: Madrid', 'Provincia: Madrid', 'NIF: 02293961X', and 'Teléfono móvil: 686506054'. The background shows the same 'GESTIÓN DE PETICIONES' interface as in the previous screenshot.

Ilustración 126: Resultado verCV asociado

Si pinchamos en el botón "Quitar", el CV asociado desaparecerá, al mismo tiempo que los seguimientos que tenga.

Por tanto, vamos a crear primero un seguimiento para "Sonia" pinchando en "Añadir Seguimiento 1".

Ilustración 127: Resultado crear seguimiento

Se abre un espacio para poder rellenar, y al final tenemos un select, cuyos valores sólo serán de los CV que haya asociados. Se pueden crear tanto seguimientos como sean necesarios. Cuando hayamos terminado todo, podremos guardar los cambios en el botón "Guardar cambios". Vamos a comprobar por último la funcionalidad del botón "Quitar". En este momento, que tenemos dos CVS asociados y un seguimiento a uno de ellos, vamos a eliminar a "Sonia" como CV asociado.

Ilustración 128: Resultado eliminar CV asociado-1

Se nos mostrará un mensaje para advertirnos de que es posible que haya seguimientos sobre ese CV, y que en ese caso se perderán.

Si le damos a "Aceptar":

The screenshot shows the 'GESTIÓN DE PETICIONES' (Request Management) interface. At the top, there is a header bar with the title 'GESTIÓN DE PETICIONES' and navigation buttons: 'Atrás', 'Inicio', and 'Cerrar sesión'. Below the header, there is a 'Guardar cambios' (Save changes) button. The main content area is divided into three sections: 'PETICIÓN', 'SEGUIMIENTO', and 'CV ASOCIADOS'. The 'PETICIÓN' section shows a request with the name 'Candidatos que sepan HTML' and the status 'abierta', with a 'Ver' button. The 'SEGUIMIENTO' section has an 'Añadir Seguimiento 1' button. The 'CV ASOCIADOS' section shows a list of associated CVs, with the first one being 'Sergio Jiménez' associated on '16 Junio 2016'. There are 'Ver' and 'Quitar' buttons for each CV, and an 'Agregar CV' button at the bottom.

Ilustración 129: Resultado eliminar CV asociado-2

Finalmente nos quedamos sólo con un CV asociado. Por último, volvemos a la página de inicio y pinchamos en la última opción que nos queda, en el botón "mantenimiento".

The screenshot shows the 'MANTENIMIENTO' (Maintenance) interface. At the top, there is a header bar with the title 'MANTENIMIENTO' and navigation buttons: 'Atrás', 'Inicio', and 'Cerrar sesión'. Below the header, there is a list of system modules: 'Usuarios', 'Idiomas', 'Provincias', 'Conocimientos', 'Titulaciones', and 'Perfiles'. Each module is represented by a blue button.

Ilustración 130: Resultado mantenimiento

Esta página muestra los distintos módulos que se pueden administrar, y a los que tendrá solo acceso un usuario administrador del sistema.

Todos funcionan igual, y si por ejemplo pinchamos en "Conocimientos":



Ilustración 131: Resultado mantenimiento conocimientos

En esta página podremos eliminar un conocimiento, editarlo, o añadir uno nuevo. Esta página servirá para tener controlada la aplicación, poder dar de alta usuarios, modificarlos, poder añadir conocimientos, eliminarlos...y todo tipo de administración que se desee.

Ya se ha mostrado el resultado de lo que es capaz de hacer la aplicación. Sin embargo, vamos a mostrar ahora la verdadera utilidad, y es la carga masiva de CVS. La idea es que se disponga de grandes cantidades de CVS en formato html, y se puedan insertar de golpe en la aplicación. El tiempo que se ahorra es muy elevado. Por ello, vamos a seleccionar por ejemplo 5 CVS:

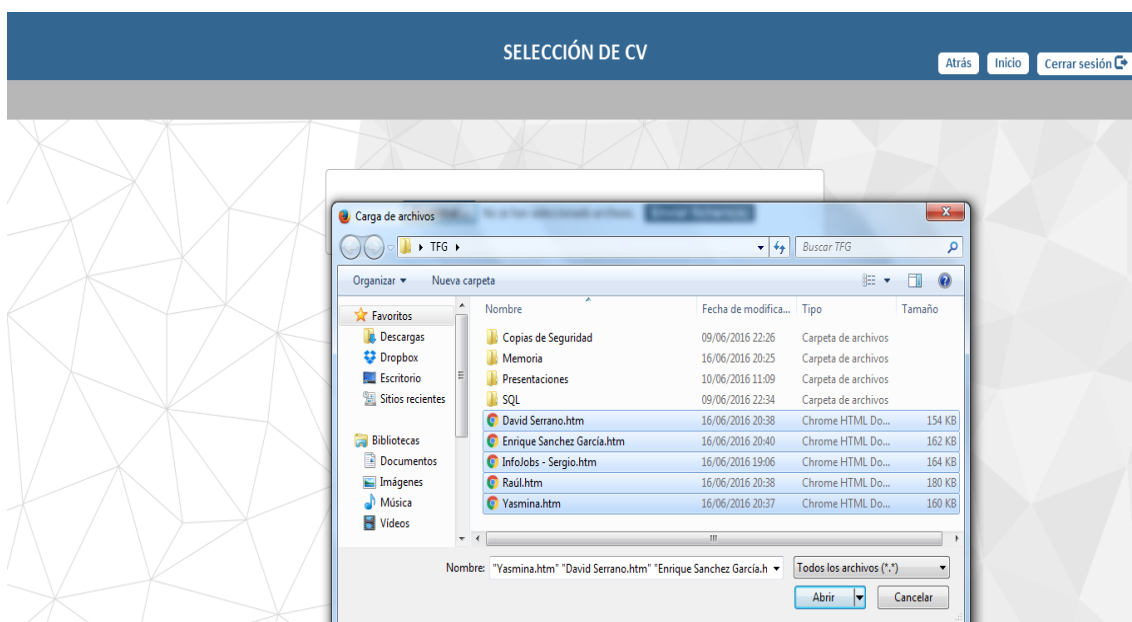


Ilustración 132: Importar varios CVS-1

Seleccionamos los 5 y le damos a enviar fichero:

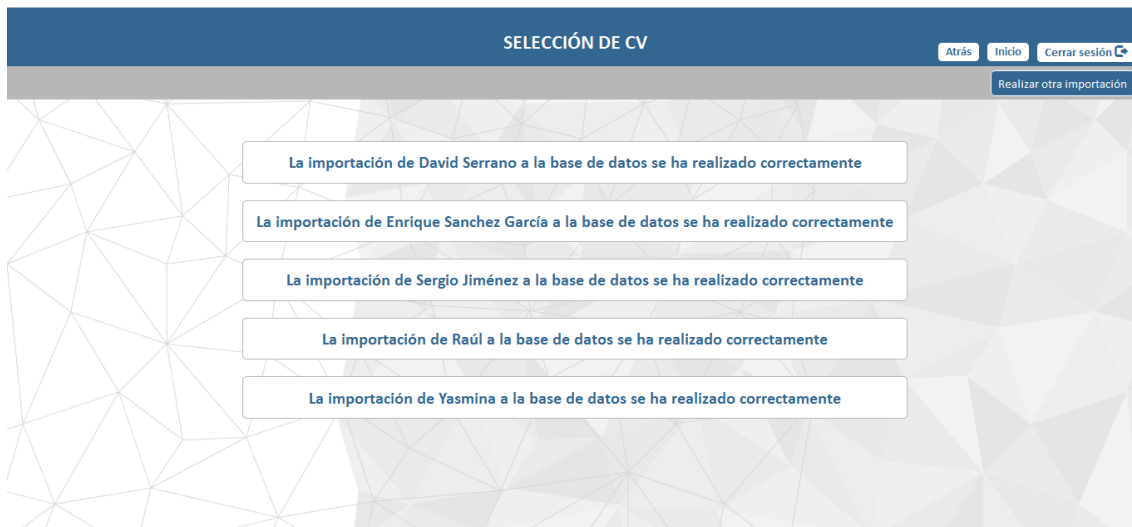


Ilustración 133: Importar varios CVS-2

Vemos como se importan correctamente todos los CVS.

7. Conclusiones

En este apartado se van a detallar las conclusiones acerca de la realización del proyecto.

Encontrar empleo es una de las mayores preocupaciones de las personas en la sociedad de hoy en día. Para tener opciones de conseguir un trabajo, hacer llegar los CVS a los destinatarios es la clave principal. Portales web se dedican expresamente a esta comunicación entre el candidato y el expositor de una oferta, tales como Infojobs o Tecnoempleo. Estos CVS, que son enviados a través de los portales web, acaban llegando a sus destinatarios, y en particular a las empresas. De aquí surge el SGCV. Miles de CVS llegan a los departamentos de selección de las empresas, y tener que manejar toda esa información es un proceso costoso. Muchas ventanas abiertas en el ordenador para ver la información de muchos CVS, o mucho papel gastado si se imprimen, además que se pueden perder o no están seguros. El SGCV facilita una solución a estos problemas.

A lo largo del documento, se ha expuesto la posibilidad de recoger un CV de un portal web, y como se ha importado directamente mediante un botón al SGCV. Más tarde, de manera clara y sencilla a través de la página web, se ha podido acceder a la información de este CV y modificarla. Por tanto, se ha cumplido el objetivo principal. Se ahorrará una enorme cantidad de tiempo con el hecho de tener una aplicación con toda la información guardada que puede ser consultada en un instante y desde cualquier sitio con acceso a internet. Además, si la información no está guardada, se puede importar el CV si se tiene en el formato, y si no incluirlo manualmente. Digamos que el SGCV facilita el trabajo a las empresas, y con ello, se favorece tanto a la empresa como a los candidatos que se inscriben en los portales web.

Un candidato puede inscribirse en un portal web, y su CV acaba llegando a una empresa. La empresa, ve su CV y lo deja como pendiente porque le ha interesado. Después, no recuerda el nombre de la persona, o no recuerda donde lo ha guardado, o cualquier otro problema. Por tanto, la oportunidad de este candidato de tener una entrevista con esa

empresa se esfuma. Con el SGCV no ocurrirá eso. El CV llegará a la empresa y se incluirá directamente al SGCV. Más tarde, cuando se quiera consultar, se realizará una búsqueda hasta encontrarlo, por ejemplo introduciendo como búsqueda el usuario de la aplicación que haya introducido CV. De esta manera, todo usuario sabrá que CVS ha subido a la aplicación, y podrá acceder a ellos fácilmente. Por tanto se cumple con esto otro de los objetivos, y es que se facilita la posibilidad de encontrar trabajo a los candidatos.

Además, se ha utilizado esta información guardada con fines muy útiles. Se ha dado la posibilidad de exportar la información a Word, lo que permite tener el CV en papel por si se quiere llevar a algún lado. Por otro lado, el módulo de peticiones permite llevar un control sobre los candidatos más interesantes para la empresa, ya que cumplen los requisitos que se exigen en la petición.

En definitiva, se han cumplido con éxito los principales objetivos que se querían abordar con el proyecto, consiguiendo una aplicación sólida, fiable, capaz de almacenar gran cantidad de información y cuya funcionalidad facilita mucho el trabajo a los departamentos de selección de candidatos de las empresas.

7.1. Trabajos futuros

En este apartado vamos a realizar unos breves comentarios acerca de posibles trabajos futuros sobre este proyecto. Aunque se ha intentando cubrir las mayores funcionalidades posibles, hay algunas que no han sido posibles y serían muy interesantes:

- **Otros portales web:** no sólo existen Infojobs y Tecnoempleo. Para el futuro sería muy interesante y útil que el SGCV pudiera recibir CVS de otros portales web, cuantos más mejor.
- **Telerik:** Telerik es una empresa que facilita el desarrollo de software poniendo al mercado productos en forma de herramientas, controles, etc. Entre otros, dispone de una herramienta capaz de almacenar grandes cantidades de información, ordenadas en tablas y de una manera vistosa y simple. Sería interesante mostrar los CVS usando Telerik. [12].
- **Formatos de exportar:** no sólo existe Word. Se podría estudiar el exportar a otros formatos como Excel o PDF, además de que el contenido que se exporta sigue un formato definido, y se podrían crear muchos más. Muchos formatos de salida de información en muchos formatos de extensión de documento.
- **Información adicional:** quizá sería interesante guardar de alguna manera información adicional sobre un candidato. Después de una llamada por teléfono o de una entrevista, el CV seguirá en el sistema, y puede ser útil apuntar información adicional extraída de esas conversaciones.
- **Otros:** con el uso se irán viendo que cosas se pueden modificar o no. Es posible que se cambien los formularios de alta, los filtros de búsqueda, se quiera añadir información a las peticiones, etc. Serían cosas que habría que ver después de un tiempo usando la aplicación.

8. Bibliografía

En este apartado se incluirán los elementos que han sido consultados como apoyo para la realización de este proyecto:

[1] Lluís Codina, "Qué es un sistema de gestión documental ", *El profesional de la información*, Año de publicación 1993, en línea. Disponible en: http://www.elprofesionaldelainformacion.com/contenidos/1993/mayo/qu_es_un_sistema_de_gestin_documental.html [consulta: 7 de Junio de 2016].

[2] Portal web Athento.com. "Gestión documental inteligente", *Athento.com*, en línea. Disponible en: <http://www.athento.com/gestion-documental-inteligente>. [Consulta: 7 de Junio de 2016].

[3] Portal web Docuware.com. "Las ventajas de los documentos digitales", *Docuware.com*, en línea. Disponible en: <https://www.docuware.com/es/Gestion-documental-soluciones/c%C3%B3mo-sacar-partido-de-los-sistemas-de-gesti%C3%B3n-documental>. [Consulta: 7 de Junio de 2016].

[4] Saul Equihua. "Data & Text Mining", *Infotecarios.com*, en línea. Disponible en: <http://www.infotecarios.com/data-text-mining/>. [Consulta: 7 de Junio de 2016].

[5] Marlon Ruiz. "Introducción a los Sistemas de Base de Datos", *Monografías.com*, en línea. Disponible en: <http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml>. [Consulta: 7 de Junio de 2016].

[6] UPCI (alias). "Diferencias entre base de datos relacional y no relacional", *slideshare.net*, en línea. Disponible en: <http://es.slideshare.net/rafq007/diferencias-entre-base-de-datos-relacional-y-no-relacional>. [Consulta: 7 de Junio de 2016].

[7] William Díaz Sepúlveda. "Bases de datos No-SQL: Llegaron para quedarse", *basesdedatosnosql.blogspot.com.es*, en línea. Disponible en: <http://basesdedatosnosql.blogspot.com.es/>. [Consulta: 7 de Junio de 2016].

[8] Jefatura del Estado, "Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal", 1999.

[9] Portal web Infojobs. "Condiciones legales de Infojobs", *Infojobs.net*, en línea. Disponible en: <https://www.infojobs.net/lex.xhtml?dgv=207005835234262024#legalem>. [Consulta: 15 de Junio de 2016].

[10] Portal web Tecnoempleo. "Política privacidad", *Tecnoempleo.com*, en línea. Disponible en: <https://www.tecnoempleo.com/politicaymarco2.php>. [Consulta: 15 de Junio de 2016].

[11] Portal web Agencia española de protección de datos. En línea. Disponible en: <https://www.agpd.es/portalwebAGPD/index-ides-idphp.php>. [Consulta: 15 de Junio de 2016].

[12] Portal web Telerik. En línea. Disponible en: <http://www.telerik.com/kendo-ui/grid>. [Consulta: 18 de Junio de 2016].

9. Anexos

En este apartado se van a incluir varios apartados como la planificación y presupuesto del proyecto, la responsabilidad jurídica del mismo y la demostración de las habilidades en inglés con el resumen de 10 páginas.

9.1. Planificación

En este apartado se explicará la planificación estimada para el desarrollo del proyecto. Primero se describirán las tareas de las que se compone el proyecto, y posteriormente se estudiarán desglosadas en un diagrama de Gantt.

Análisis

La primera tarea es la de análisis. En esta tarea se definirá el problema a solventar. Se hará un estudio de cuál será la funcionalidad del sistema, pero no que vías hay que utilizar para alcanzar esa funcionalidad. Esta tarea forma parte de un proceso iterativo e incremental, lo que quiere decir que según avancen las tareas y el desarrollo del proyecto, existe la posibilidad de volver a esta tarea y adaptarse a posibles cambios.

Diseño

La segunda tarea será la de diseño. Una vez conocido el problema a afrontar, se estudiarán las vías para su resolución. Habrá varias alternativas que estudiar hasta optar por una solución final. Esta tarea forma parte de un proceso iterativo e incremental, lo que quiere decir que según avancen las tareas y el desarrollo del proyecto, existe la posibilidad de volver a esta tarea y adaptarse a posibles cambios.

Implementación y pruebas

La tercera tarea será la de implementación y pruebas. Una vez visto el problema a resolver y las vías de cómo hacerlo, se emplearán las herramientas y tecnologías necesarias para su resolución. Esta tarea se puede dividir en sub tareas, que son:

- Creación de la base de datos: según avance la implementación y el desarrollo del proyecto, la base de datos puede cambiar constantemente hasta tener una versión final.
- Creación del programa de recogida de información: de la misma manera, según avance el desarrollo se puede necesitar una información u otra, por lo que el programa puede cambiar constantemente hasta una versión final.
- Creación de la página web: dependiendo de la funcionalidad que se quiera añadir, se puede cambiar la página hasta una versión final.
- Pruebas: a lo largo de toda la implementación, se irán realizando pruebas que validen el funcionamiento que se pide.

La fase de implementación es dependiente de la fase de pruebas, y viceversa. Cada tarea de implementación lleva asociado una serie de pruebas para comprobar la funcionalidad. Por tanto, para garantizar un sistema fiable y funcional al 100%, se contará con un colchón de tiempo para comprobar que todo funciona correctamente.

Diagrama de Gantt

Una vez descritas las diferentes tareas del proyecto y como se conectan entre sí, en este apartado se va a mostrar el diagrama de Gantt del proyecto, que muestra de manera visual el tiempo previsto para cada una de las tareas que componen el proyecto.

En primer lugar aquí está la tabla de actividades detallada:

	Actividad	Fecha de inicio	Duración (días)	Fecha de fin
Análisis	1	14/09/2015	25	09/10/2015
Diseño	2	09/10/2015	20	29/10/2015
Creación de base de datos	3	29/10/2015	5	03/11/2015
Creación de programa de recogida de información	4	03/11/2015	10	13/11/2015
Creación de página web	5	13/11/2015	35	18/12/2015
Pruebas	6	15/11/2015	60	14/01/2016

Tabla 89: Actividades de planificación

En esta tabla se muestra lo siguiente:

- Lista de actividades numeradas del 1 al 6.
- Fecha de inicio de cada una de las actividades.
- Duración, en días, de cada actividad.
- Fecha de fin de cada una de las actividades.

A continuación, se muestra el diagrama de Gantt:

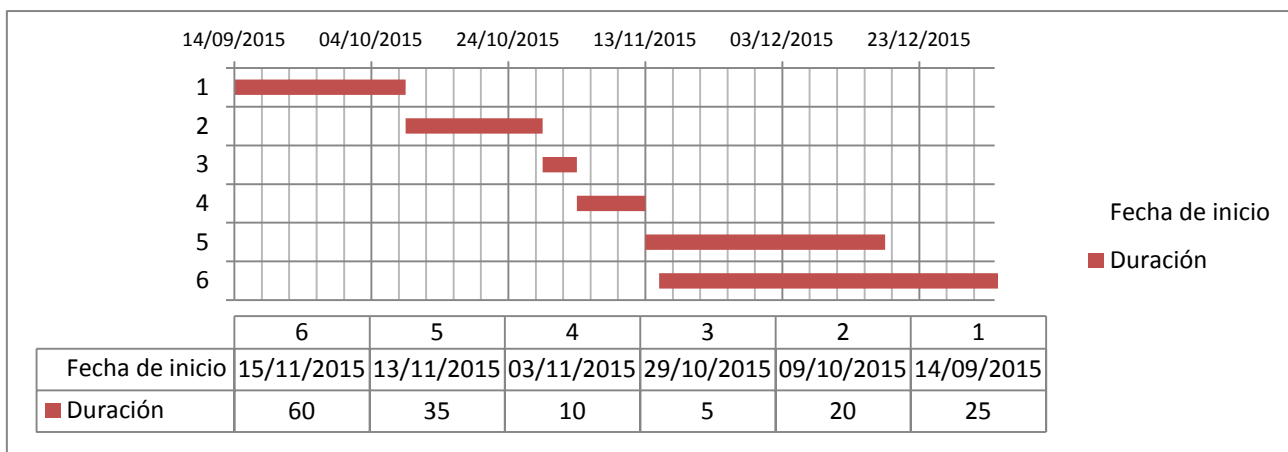


Ilustración 134: Diagrama de Gantt

En el diagrama aparece lo siguiente:

- En la parte superior se pueden ver las fechas que sirven como referencia.
- Del 1 al 6 son las distintas actividades explicadas anteriormente.
- Las barras rojas hacen referencia a la duración de cada actividad.
- La tabla inferior muestra una pequeña leyenda de la duración (en días) de cada actividad, con su fecha de inicio.

Al disponer sólo de un informático para realizar el proyecto, las tareas no pueden solaparse, o al menos deberían solaparse lo menos posible. En este caso, primero se harían las tareas una tras otra y en orden cronológico según lo establecido en el diagrama. No obstante, la fase de pruebas se solapa con la de creación de la página web. Esto es porque una vez está muy avanzado el sistema y se está creando la página web donde se tiene que ver todo, es conveniente ir realizando pruebas poco a poco para ir comprobando el funcionamiento.

Además, el orden planificado sería el mostrado en el diagrama, pero durante su desarrollo puede variar. Es posible que durante la creación de la página web, haya que volver a tomar una decisión de diseño, o cambiar la base de datos. Digamos que las tareas pueden requerir revisarlas aún estando digamos finalizadas. Vemos además como se deja un margen de tiempo al final, que sirve para realizar pruebas a fondo, ultimar algunos cambios y comprobar la viabilidad y funcionamiento total del sistema.

Fecha de inicio estimada	14/09/2015
Fecha de fin estimada	14/01/2016
Duración total (días)	111

Tabla 90: Duración total del proyecto

Dado que la fecha de inicio estimada del proyecto es el 14/09/2015 y la fecha de fin el 14/01/2016, la duración del proyecto total se estima en: **122** días naturales.

9.2. Presupuesto

En este apartado se va a detallar el presupuesto necesitado para poder llevar a cabo el proyecto de principio a fin. El presupuesto total viene determinado por lo siguiente:

Coste de material Hardware

Para la realización del proyecto ha sido necesario el empleo de un material hardware específico, el cuál ha sido detallado en el apartado 2.3. Medios hardware. El coste viene detallado en la siguiente tabla:

Material	Coste
Ordenador*	33,33 €
TOTAL	33,33 €

Tabla 91: Presupuesto material hardware

**El ordenador es en concreto un ordenador portátil. La vida útil de un ordenador portátil de estas características oscila entre los 4 y 6 años, pongamos 5. La amortización anual de un 20 % serían 100 euros al año, y la duración del proyecto son 4 meses, por tanto una amortización del 6,6%, a 33,33 €.*

Coste de material Software

Para la realización del proyecto ha sido necesario el empleo de un material software específico, el cuál ha sido detallado en el apartado 2.3. Medios software. El coste viene detallado en la siguiente tabla:

Material	Coste
Sistema operativo (Windows 7) *	0 €
XAMPP **	0 €
Notepad++ **	0 €
Licencia Microsoft SQL Server 2012 ***	0 €
Navegador Mozilla firefox **	0 €
TOTAL	0 €

Tabla 92: Presupuesto material software

** El coste viene incluido en su coste hardware (Ordenador).*

*** XAMPP, Notepad++ y Mozilla firefox son de descarga gratuita.*

**** La universidad Carlos III proporciona licencias gratuitas a los estudiantes.*

Coste de mano de obra

La mano de obra viene determinada por el trabajo un ingeniero informático.

Rango	Horas trabajadas	Total
Consultor (ingeniero informático)	40 horas semanales * 16 semanas * 12 €/hora	7.680 €
	TOTAL	7.680 €

Tabla 93: Presupuesto mano de obra

Coste total del proyecto

El coste total del proyecto será la suma de los costes derivados explicados anteriormente. Además, se añade un presupuesto añadido ("colchón") por si surge cualquier imprevisto a lo largo del desarrollo del proyecto. Este resultado se muestra en la siguiente tabla:

Material Hardware	33,33 €
Material software	0 €
Mano de obra	7.680 €
Presupuesto añadido	2.286,67 €
TOTAL	10.000 €

Tabla 94: Presupuesto coste total del proyecto

9.3. Responsabilidad jurídica

En este apartado se expondrán las responsabilidades legales que supone el manejo de los CVS de los candidatos. Esta responsabilidad, es única por parte del SGCV, sin embargo viene precedida por dos vías, que son los portales web donde se obtienes los CVS, Infojobs y Tecnoempleo.

Infojobs

Todo candidato que utiliza Infojobs, y accede a él creándose una cuenta propia y elaborando un CV con sus datos personales, está condicionado por las "Condiciones legales de Infojobs" [9].

De entre toda la documentación, destaca lo siguiente:

"Infojobs condiciona la utilización de la mayoría de sus servicios a la previa cumplimentación del correspondiente registro de Particular al Sitio Web de conformidad con la Ley 15/99 de Protección de Datos".

"Infojobs es consciente de que para los Particulares es sumamente importante que se dé un tratamiento responsable en la gestión de sus datos personales. Por ello Infojobs cumple todas las normas legales de protección de datos que sean aplicables y establece mecanismos y procedimientos de seguridad para la gestión de estos datos".

"Si el Particular decide registrarse en Infojobs, se solicitarán los datos necesarios para la consecución de los fines y finalidades a los que está destinado el Sitio Web. De conformidad con lo establecido en la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal, se informa al Particular que los datos personales que facilite, incluidos los datos curriculares, serán incorporados a un fichero registrado ante la Agencia Española de Protección de Datos, cuyo titular es Schibsted Classified Media Spain, S.L.".

"Infojobs permite crear páginas dedicadas a una marca, negocio y misión a empresas, a universidades, a organizaciones sin ánimo de lucro y a otras organizaciones así como a expertos reconocidos y a personas influyentes. Estas páginas son públicas y la entidad que las creó podrá acceder a cualquier comunicación e información compartida a través de ellas".

"Al rellenar el primer formulario y hacer clic para enviar los datos, el Particular manifiesta haber leído y aceptado expresamente la presente política de privacidad incluidas dentro de las condiciones legales del Particular, y otorga su consentimiento inequívoco y expreso al tratamiento de sus datos personales conforme a las finalidades informadas y servicios que presta Infojobs, así como la cesión de sus datos personales a terceros para el cumplimiento del servicio (con los condicionantes que se especifican en el apartado de control de privacidad). A estos efectos se entiende por Tercero 'las entidades, Organizaciones que utilizan los servicios de Infojobs para la búsqueda y selección de personal' ".

Infojobs deja claro que informa al usuario de todas las condiciones de uso del portal, de cómo se va a tratar la información que cada candidato incluya, y todo ello regido por las leyes de protección de datos incluidas en el BOE.

Tecnoempleo

Todo candidato que utiliza Tecnoempleo, y accede a él creándose una cuenta propia y elaborando un CV con sus datos personales, está condicionado por la "Política de privacidad" [10] de Tecnoempleo.

De entre toda la documentación, destaca lo siguiente:

"Esta sociedad es titular del Portal de Empleo TECNOEMPLO®, y de su Base de Datos inscrita en la Agencia de Protección de Datos con el Código 2000410141, y se ajusta a la normativa establecida en la Ley Orgánica 15/1999, de 13 de Diciembre, de Protección de Datos de Carácter Personal (BOE de 14/12/1999)".

"Los usuarios (empresas y candidatos), al pulsar en el formulario de Alta el botón "ALTA EMPRESA/ALTA CANDIDATO", dan su expreso e inequívoco consentimiento al tratamiento de los mismos, así como, en su caso, a la cesión de los mismos a terceros para el cumplimiento del servicio, dentro del marco establecido por la legislación vigente".

"Los candidatos tienen la posibilidad de mantener sus datos personales en la privacidad, dejando sólo manifiestos los pertenecientes a su perfil profesional. Si, una vez localizados éstos por las empresas autorizadas éstas desearan ponerse en contacto con tales candidatos, Tecnoempleo® solicitará cada uno la autorización para facilitar dicho contacto entre la empresa concreta y el candidato, pudiendo rechazar esta posibilidad".

"Los datos son tratados por aquellas Empresas oferentes a las que voluntariamente el profesional envía su demanda de empleo, y por las autorizadas expresamente por Tecnoempleo® para consultar su base de datos, pudiendo, en este caso, proteger sus datos personales a través del nivel de privacidad de su currículum".

Tecnoempleo deja claro que informa al usuario de todas las condiciones de uso del portal, de cómo se va a tratar la información que cada candidato incluya, y todo ello regido por las leyes de protección de datos incluidas en el BOE.

Responsabilidad jurídica del SGCV

Como bien se detalla en las condiciones legales de ambos portales web, el candidato al darse de alta, acepta entre otras cosas:

- Cesión de sus datos personales a terceros para el cumplimiento del servicio del portal web (Infojobs o Tecnoempleo).
- Se entiende por tercero aquellas organizaciones que utilizan los servicios de estos portales para la búsqueda y selección de personal.

El SGCV ha sido un proyecto desarrollado dentro de una organización con acceso específico a los portales web de Infojobs y Tecnoempleo, por lo que el acceso a esa información está dentro de la legalidad. En principio, el uso del SGCV será interno y propio para la organización, por lo que nadie externo a la organización podrá acceder a la información personal de los candidatos.

Sin embargo, en caso de que en algún momento, la organización encargada de realizar el SGCV quisiera hacer accesible la información o publicarlos en algún medio, sería necesario cumplir como mínimo con las mismas exigencias legales, de Infojobs y Tecnoempleo, a la hora de tratar la información personal de los candidatos.

En este caso, la organización se vería obligada a cumplir con la "**Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal**" [9]. La LOPD establece la exigencia de que cuando se proceda a la captación de datos personales, se deberá notificar la existencia del fichero a la Agencia Española de Protección de Datos [11] indicando una serie de detalles como el responsable del fichero, la finalidad, su ubicación, etc.

De este modo, si la organización que ha realizado el SGCV facilita los datos de los CVS de Infojobs o Tecnoempleo a un tercero, dicha normativa obliga a la organización a proteger estos datos de un uso fraudulento o lo que es lo mismo, que dicha organización no comercie con los datos y que los tenga debidamente protegidos.

10. Habilidad en Inglés

En este apartado se va a incluir un resumen en inglés del proyecto, tal y como se especifica en los requisitos del TFG para alumnos del plan 2011 con Grado en Ingeniería Informática, como es mi caso.

Introduction

Nowadays, we find that technology includes the entire social environment around us. At the time to find a job, people make a CV where expose their personal information, their studies, their professional experience... to get that companies can be interested in hiring them.

Recently, everybody used to give the CV in person to the companies or individuals who performed an offer. However, with the progress of new technologies: internet, computers, mobile phones...these CVS are sent by Internet, often via email.

The CVS management is a very important issue, most of the people hired had a good organized and concise CV. This system helps in the selection process of CVS in some web portals widely used.

The companies have a department of selection, which are dedicated to see CVS that get certain requirements and they contact with these people. The person who is responsible for carrying out this process may be interested in sending those same CVS to other companies or other sectors that might need these profiles, or keep them for themselves to manage it in the future. This system will facilitate this process.

The fundamental part of the system is to send the CVS to the destinations interested in them, and to view information CV clearly and to make changes. A CV is one of the main routes to find a job, which is a priority to the most of the people. This is the main motivation for creating this system, facilitate and expedite the selection of candidates for a job vacancy.

The principle objective of this project is to make a system capable of storing CVS, to manage them and export them to a particular format. Showing how the system can store CVS in the database, either manually created or caught from web portals. An analysis will be made in depth how the registration process of CVS is performed.

On the other hand, will be described in detail the management capabilities once saved CV (modification of information, association of CVS with requests, the progress of the CV, among others). Requests may also be created, which will have a number of requirements and which can be assigned multiple CVS provided they meet all the requirements.

Finally, we will test how a CV can be exported to various formats previously created and defined.

Abstract

The abstract will focus on the most relevant sections of the project. These sections are: state of the art, analysis, development, result and planning and estimation.

State of the art

In this technological society in which we live, finding a job is a fundamental reason in the life of all people. To do this, have a good CV is essential for achieving the goal. To make the SGCV is required the creation of various elements and the use of various techniques and concepts. These are:

- Document management systems: document management system is a program that allows managing a large number of documents, so they can be stored in a database, you can access them easily and treat them as desired. In this case, the SGCV is a particular case of documentary system, specifically a type of document management software.
- Document management software: document management software is just an application that is responsible for performing all the functions of a document management system. That's the SGCV.
- Data mining: Data are the raw material, which will give meaning users into information and later the specialists transform it in knowledge.
- Text mining: Information retrieval, i.e., select the relevant texts. Removing the information included in these texts: facts, events, key facts, relations between them, etc. Finally what would be done before we defined as data mining to find associations between these previously extracted from the text key data.
- Database systems: a system database to store, modify and remove large amounts of information that have a particular relationship to each other. For the particular case of SGCV, it has been necessary to have a system database, in which you create a specific database with tables, relationships, and other components, where you can store all the information of the CVS.
- SQL vs. No-SQL: models relational databases (SQL) do not allow fault tolerance and redundancy, it seems clear that the choice for the SGCV is the model of a relational database.



Ilustración 135: State of the art

Analysis

To analyse the project, focus on the following aspects: obtaining CVS, database, information collection program, website, system requirements and uses cases.

Obtaining CVS

The main idea is to get a lot of CVS web portals and introduce them automatically slammed into the system. The goal of this programme is the time saved when having to store the information.

Therefore, we must see how these CVS are imported into the system. Selection department of the companies has special access to web portals. This access allows access to certain CVS. A search is made with certain criteria and is selected one of the relevant results. Later, it is possible to save the CV in html format. This saved file is the CV that can be imported to SGCV to save the information.

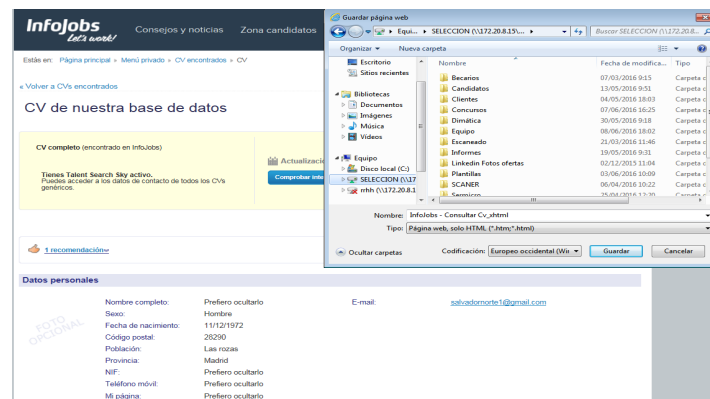


Ilustración 136: Obtaining CVS

Database

Firstly, we should create a connection between the server and database to access it and add, modify or delete information. The ODBC connector is used, introducing specific parameters to connect to the server that contains the database, so you can access it through PHP code.

Once we have the connector, we can create the database. "Microsoft SQL Server 2012" tool has been used. This tool allows you to create a database very easily. Once created, they will be included in all tables that are necessary, with their attributes, their relationships with each other...etc.

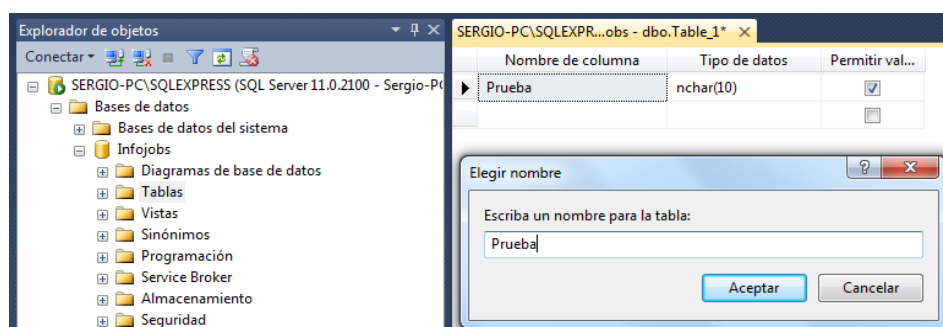


Ilustración 137: Database

Information collection program

The information collection program is designed with the PHP programming language. The program will receive a CV, saving the parameters that refer to CV and will check the format used. Depending on the web page where is saved, the system execute a function or another.

```
<?php
$myFile=$_FILES['fichero_usuario'];
$fileCount=count($myFile['name']);

for ($i = 0; $i < $fileCount; $i++) {
    $file=$_FILES['fichero_usuario']['tmp_name'][$i];
    if (is_uploaded_file($file)) {
        comprobarFormato($file);
    }
    else {
        echo "Error:".$_FILES['fichero_usuario']['error'] . " ";
    }
}

function comprobarFormato($file){
    $html = new simple_html_dom();
    $html = file_get_html($file);

    if(!is_null($variableComprobar = $html->find('title',0)->plaintext)){
        $comprobacion = strpos($variableComprobar, 'Tecnoempleo');
        if($comprobacion !== false){
            insertarTecnoempleo($file);
        }
        else{
            insertarInfojobs($file);
        }
    }
}

function insertarInfojobs($file){

}

function insertarTecnoempleo($file){

}

?>
```

Ilustración 138: Information collection program

Web page

The website has been developed with several elements. Firstly, using JavaScript programming language, this provides functionality to the page elements. It also had been used the language of html tags, which allows viewing in the browser the elements of the screen, such as a title or a button. This markup will be complemented with CSS, whose function is to provide style to those elements of the screen, such as focus buttons, or putting a background image. PHP language is embedded, allowing access to the database to gather information, or to modify it.



Ilustración 139: Web page

System Requirements

A collection of requirements describes the characteristics or attributes of the system. The requirements explain what the system has to do, not how it does. These requirements are differentiated in two groups to facilitate understanding of the system necessities: functional and nonfunctional requirements.

1- *Functional requirements*: those that describe the performance and function of the system. In turn, they are divided into several groups:

- Operation requirements: determine how the system works: workflow, functionality, etc.
- Interface requirements: They determine how the system will be displayed to the user, and how it will interact with it.
- Database requirements: determine how they will store data in the system.

2- *Nonfunctional requirements*: those that describe tasks such as maintenance, supervision, quality, safety, etc.

ID	FF01	Título	Register
Tipo	Operation	Fecha	17/06/2016
Descripción	The system must allow user registration by an "Administrator".		

Tabla 95: Example requirement

Use cases

Use cases specify the behavior of the system as well as the sequence of interactions between the system and the user. After specifying the cases of possible use, it will be created use cases diagrams. These diagrams are used to specify the communication and the system behavior related to the users. From these diagrams we obtain 4 types of users who can access to the application:

- User "Administrator": this user will be able to perform all possible actions in the system. It will be responsible for the application maintenance, upload CVS and this modification and sign in users in the system.
- User "Petition": this user can view CSV by your search, and manage and modify requests.
- User "Manager": this user can view CSV and requests through your search.
- User "External user": this user may only view CSV using your search.

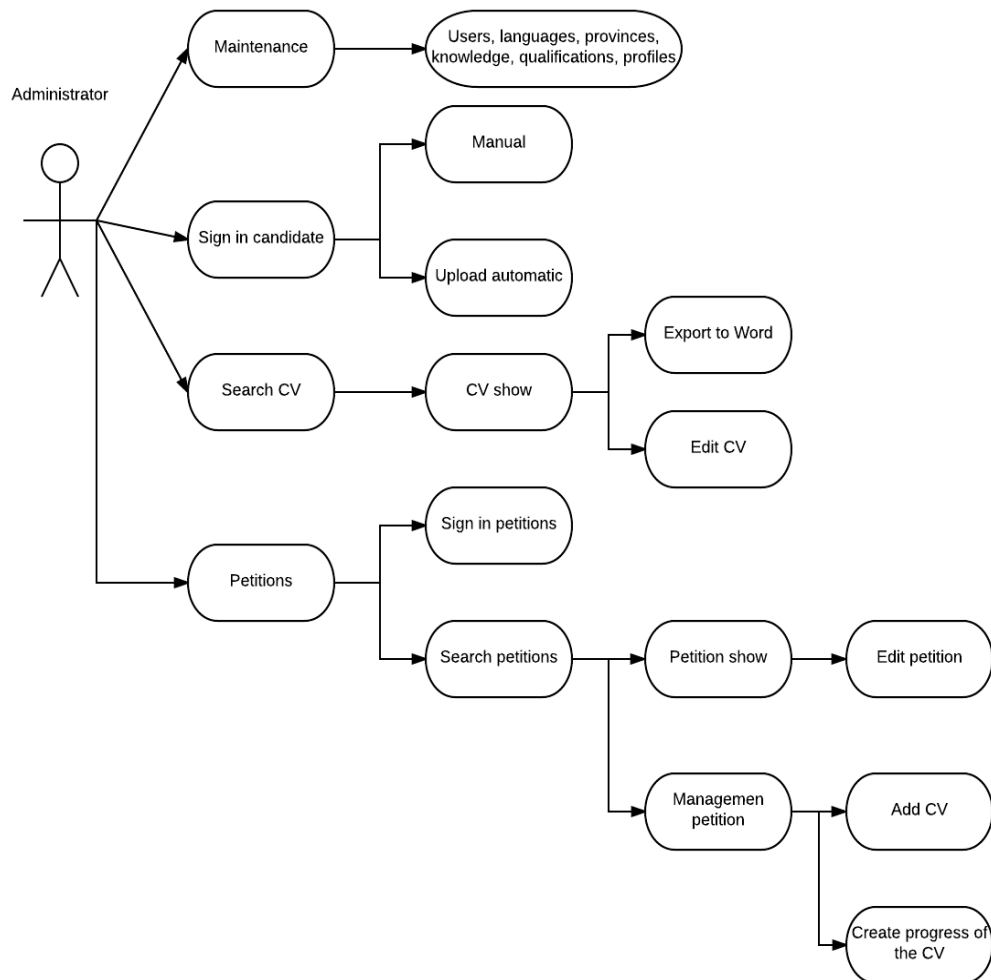


Ilustración 140: User "Administrator"

Development

To development the project, focus on the following aspects: database, information collection program and website.

Database

The information contained in the CVS has been studied, analyzing CVS included in web portals. Also, new ideas have been thought to include additional information that might be useful. In this way, I have created a relational design, with their tables, attributes, and corresponding relationships.

All tables have an identifier for each tuple, which is generated by a PHP function called NewGuid that automatically generates a guid. This will ensure that can't be repeated tuples by comparing with this guid.

There are three aspects to consider:

- Tables that don't change: there are several tables with the same purpose, with an initial insertion, and it isn't intended to change. These tables are for example qualifications, countries, provinces, etc.

- Tables that change rarely: for example conocimientos_mantenimiento table, this contains, a priori, all possible knowledge. However, someone may introduce a knowledge that doesn't exist in the table, so that the table would be modified.
- Tables "Father": these tables are Datos_personales and "Peticion". The information on a CV contains personal information, studies, professional experience, languages, etc. All this information is of the same person, so all tables inherit the identifier from Datos_personales table. The same applies to the Peticion table. CVS aggregates, the progress of the CV, the language of the request, etc., these tables will inherit the identifier of the request table.

Information collection program

The program will read a CV in html format and will save their information in the database. To do this, the website has a section to select a file, and then click on the "Send file(s)" button. This button will be associated with the program in question.

The selected CV will be received, and with the using the library "html dom", we can obtain the information of CV using html tags. Firstly, we search the word "Tecnoempleo" in the CV, searching the first tag 'title' in the document. Depending on the result, will run a function determinate (insertarInfojobs or insertarTecnoempleo).

With the same library, "html dom", we search tags like "NombreCompleto", "email", "sexo", etc., and the result is saved in variables. If there recursion, it will consider creating recursive variables. Once obtained all values, the information is inserted in the database.

If the insertion is successful, a success message is displayed. It is possible that the person's CV already has been inserted in the database. In that case, all information associated is erased, and inserted the new information.

Web page

The website will serve to display information from the CVS, and to perform a treatment of them. The sections most complex of the screens are explained, and will serve as an example of all pages.

The header, buttons, background...this things will be reused in every screen. By mixing the above-mentioned languages (PHP, JavaScript, HTML and CSS) all pages will be created.

Result

The result of the application is very visual. Using the website, we will have to import a CV. After, we will verify that this CV is in the database and can access its information.

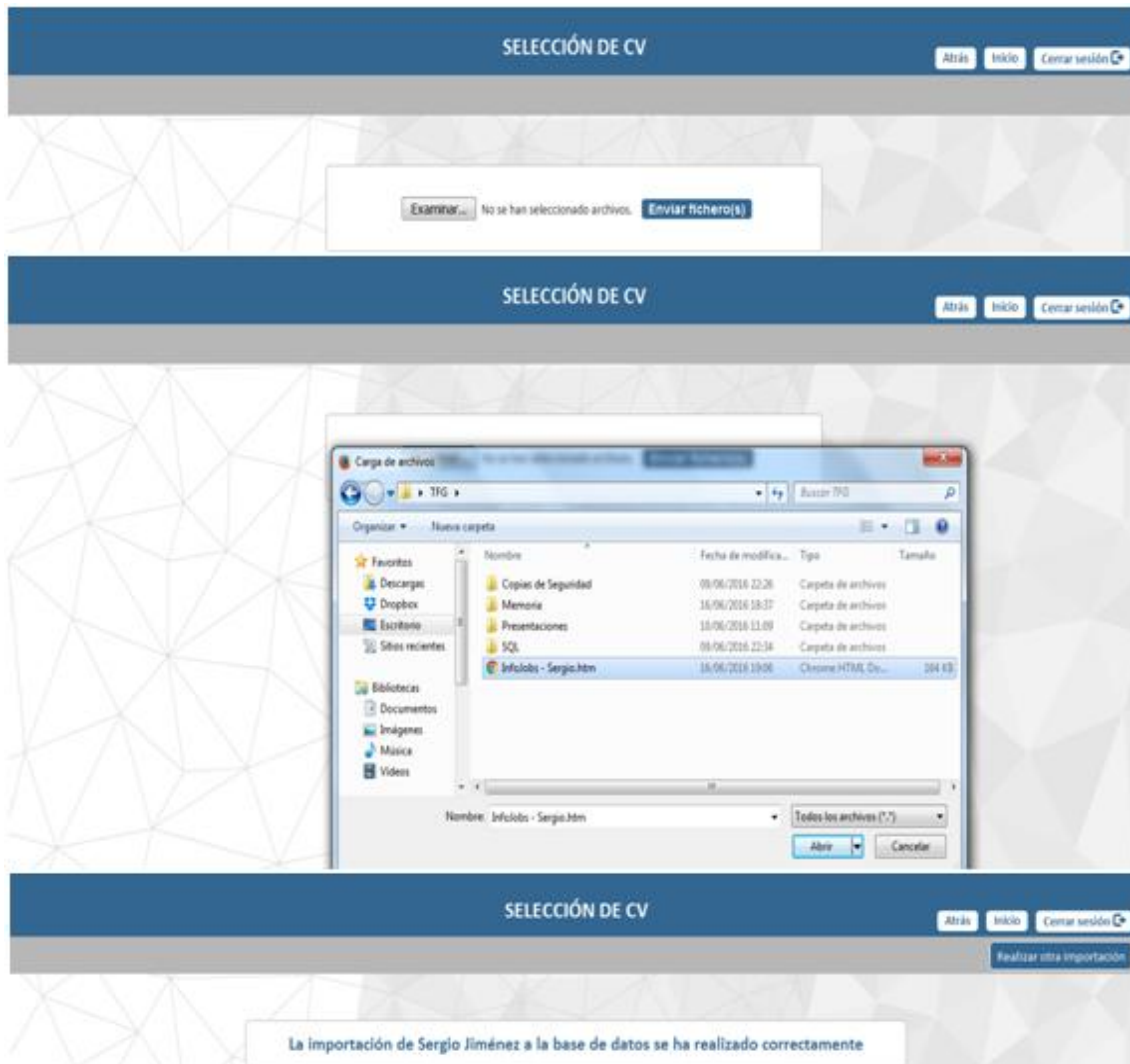


Ilustración 141: Upload CV

That is an example to import. We can see that is too easy the import CV in the application.

Now let's see if we can access to the information:

BÚSQUEDA CV [Atás](#) [Inicio](#) [Cerrar sesión](#)

Filtros

Tecnología: experiencia (años): Nombre:

Tecnología: experiencia (años): NIF:

Tecnología: experiencia (años): Empresa:

Tecnología: experiencia (años): Idioma:

Tecnología: experiencia (años): Titulación:

[Buscar](#)

Sergio Jiménez

Sonia

CV DE Sergio Jiménez [Atás](#) [Inicio](#) [Cerrar sesión](#)

Seleccione el formato del CV: [Exportar a WORD](#) [Editar datos](#)

DATOS PERSONALES

Nombre completo: Sergio Jiménez NIF: 02293961X Email: sergio.serjo@gmail.com Sexo: Hombre Fecha de nacimiento: 29/05/1992

Código postal: 28021 Población: Madrid Provincia: Madrid Teléfono móvil: 986506054

Perfil público Sergio Jimenez en InfoJobs

Mi página www.sergiojimenez.com

ESTUDIOS

Título: Grado en Ingeniería Informática Especialidad: Ingeniero en Informática Centro: Universidad Carlos III de Madrid (Leganés) Periodo: 01/2010 - 06/2016

OTROS TÍTULOS

No hay datos para mostrar

OTROS CURSOS

No hay datos para mostrar

Ilustración 142: Example of result search

The result of the search with the name "s" is "Sergio Jimenez", among others. This result is the previous import. Selecting this option, we can see the information in different sections, the option to export to "Word" and the button to edit data.

Planning and budgeting

For the project, has been necessary a preliminary planning. This planning is determined by a breakdown of activities:

	Activity	Start date	Duration (days)	End date
Analysis	1	14/09/2015	25	09/10/2015
Design	2	09/10/2015	20	29/10/2015
Creation database	3	29/10/2015	5	03/11/2015
Creation information collection program	4	03/11/2015	10	13/11/2015
Creation web page	5	13/11/2015	35	18/12/2015
Test	6	15/11/2015	60	14/01/2016

Tabla 96: Planning activities

These activities are reflected in the following Gantt chart:

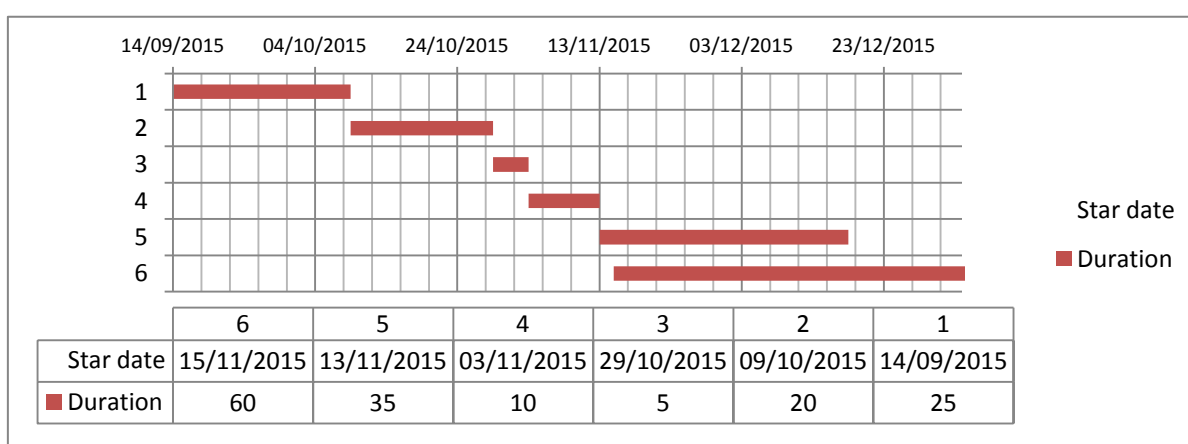


Ilustración 143: Gantt diagram

The project budget is determined by the next table:

Material Hardware	33,33 €
Material software	0 €
Labour	7.680 €
Added budget	2.286,67 €
TOTAL	10.000 €

Ilustración 144: Total budget

Conclusions

The main concern of people is to have a job. To achieve this, a fundamental part is to have a good CV. Web portals are specifically dedicated to management CV. These portals send CVS to companies that offering jobs. Companies receive many CVS, and manage them all is a costly process. The SGCv will serve to facilitate the work of CVS management.

In the document has been shown the main objective of SGCv. We can obtain a CV directly from a web portal, import the CV to the system and see your information to treat it. This simple process saves a lot of time. For candidates, these functions help them finding a job.

Also, it has shown the ability to export the information to a Word document, which is very useful. In addition, it explained the request module, which allows to manage the candidates.

In summary, we have successfully achieved the main objectives of the project, resulting in an application that simplifies the work to workers of companies whose function is the selection of candidates.

Future work

Although it has been trying to cover the greatest possible functionality, there are some that have not been possible and would be very interesting:

- **Other web portals:** be able to obtain CVS from other web portals.
- **Telerik:** test the Telerik software tool to display the CVS.
- **Format to export:** besides Word, include more formats such as PDF or Excel.
- **Additional information:** include a section for additional information to fill after an interview with the candidate.
- **Others:** other features with the use the system.